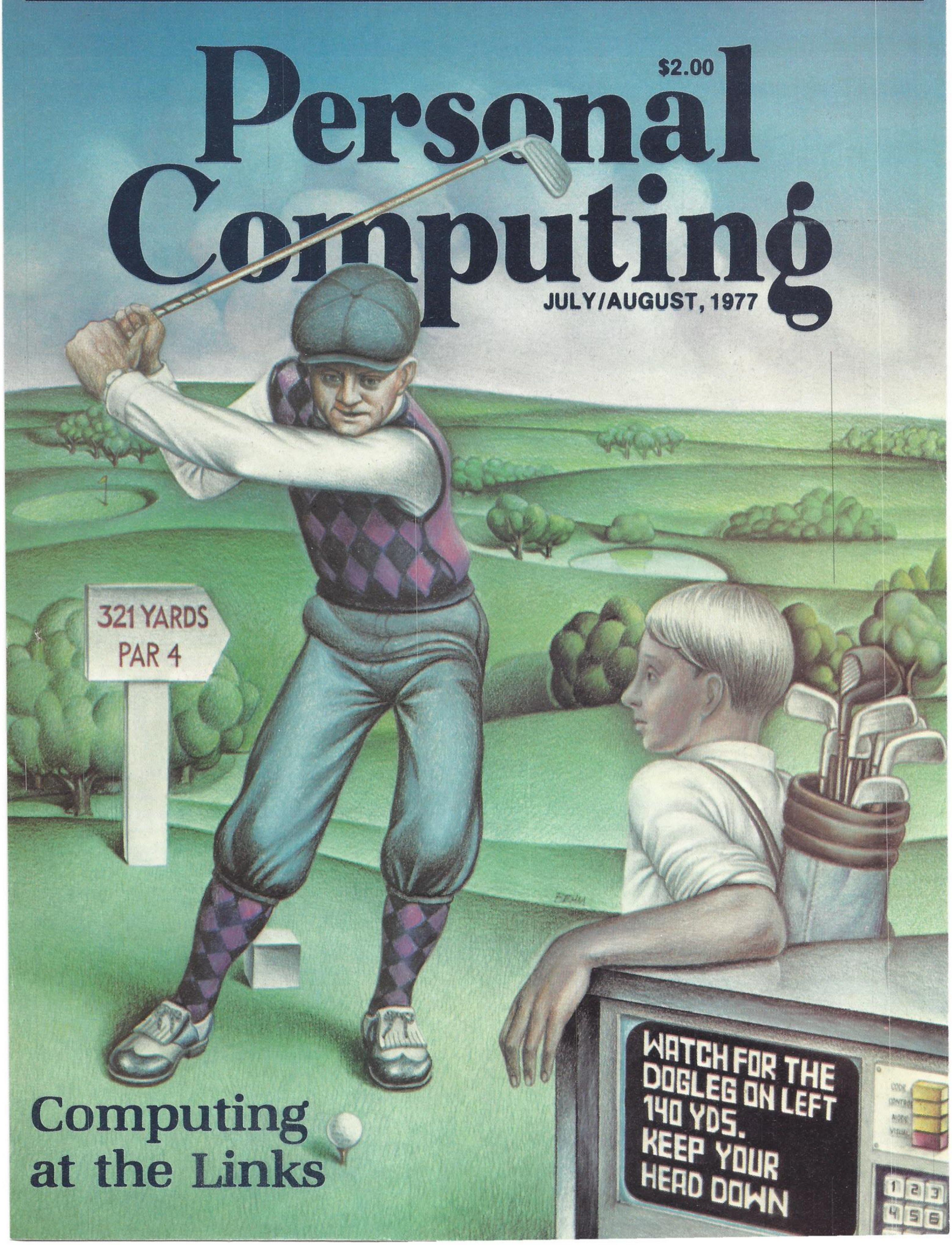
Personal Computing for Teachers • Tax Aspects of Home Computing

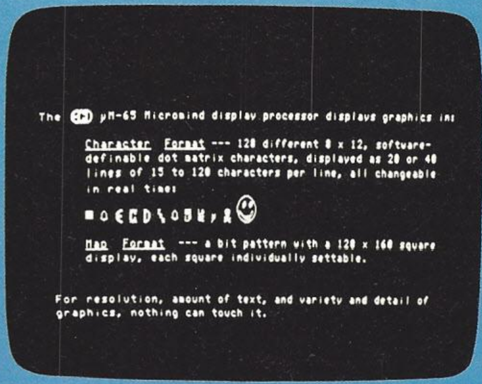
Personal Computing

$2.00

JULY/AUGUST, 1977



Computing at the Links



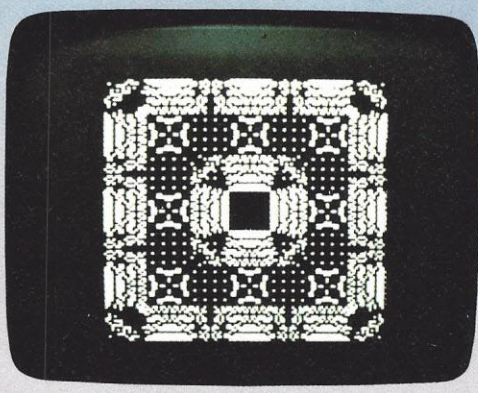
Key Into Maxi-Power @ Micro-Price

Micromind is an incredibly flexible, complete and expandable, hardware/software, general purpose computer system. You won't outgrow it.

Hardware includes an 80 key, software-definable keyboard, I/O interface board, 6500A-series microprocessor (powerful enough for advanced computing), a high-detail graphics and character display processor, power supply, rf modulator, and connections for up to 4 tape recorders plus TV or monitor. An interconnect bus permits 15 additional microprocessors, parallel processing and vastly increased computing power.

System software—including ECD's own notsoBASIC high level language, on advanced error-correcting tape cassettes—provides a word processing editor, a powerful assembler, a debugger, a file system, graphic routines, and peripheral handlers. We also include dynamic graphic games: Animated Spacewar and Life.

ECD's standard Micromind µM-65 supplies 8K bytes of memory. Additional 32K byte expansion boards and a mapping option give Micromind expandable access to 64 Megabytes. Utilizing software-controlled I/O channels, Micromind's advanced encoding techniques load data from ordinary tape recorders at 3200 bits per second.



Micromind comes to you ready-to-use, factory assembled and fully tested. Among microcomputers, it has the largest memory capacity and the fastest storage. You're looking at the work of the finest display processor on the market. You won't find a microcomputer with a more powerful CPU. You won't find a computer with a more flexible keyboard. You won't find anything to touch it at **$987.54.**

So, quit the kluge scene and key into Micromind. You'll be a main frame performer, with all the comforts of home. We're not fooling...this is the cat's µ!

ECD CORP.
196 Broadway, Cambridge, Mass. 02139
(617) 661-4400

ECD

Name ______________________

Address ______________________

City/State ______________ Zip ________

☐ Fantastic! Check enclosed: **$987.54.** Shipping paid by ECD

☐ BankAmericard ☐ Master Charge Mass. Resident add 5% Sales Tax

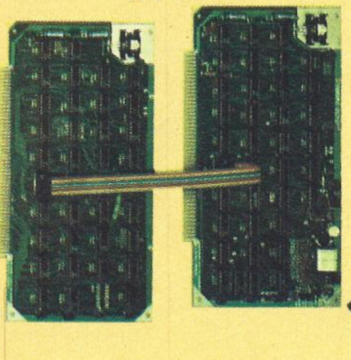
\# ______________ Expiration Date ________

Signature ______________________

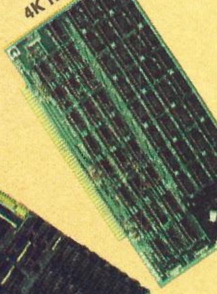
☐ Send me your brochure.

P

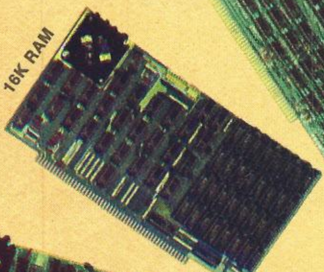
TV DAZZLER



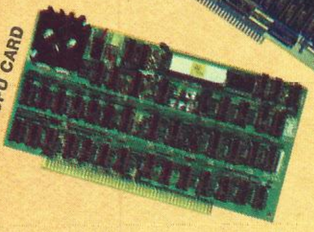
4K RAM



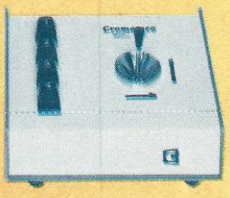
16K RAM



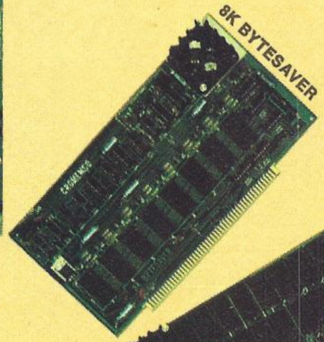
CPU CARD



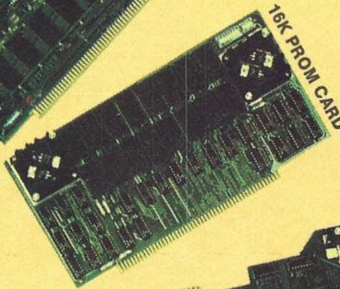
JOYSTICK CONSOLE



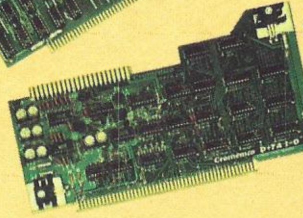
8K BYTESAVER



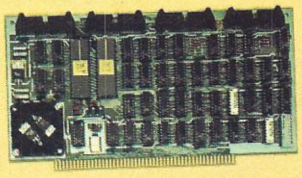
16K PROM CARD



ANALOG I/O



TU-ART



To make your computer more useful– a wide choice of memory, I/O, CPU

Your computer's usefulness depends on the capability of its CPU, memories, and I/O interfaces, right?

So here's a broad line of truly useful computer products that lets you do interesting things with your Cromemco Z-1 and Z-2 computers. And with your S-100-compatible Altairs and IMSAIs, too.

CPU

- Z-80 MICROPROCESSOR CARD. The most advanced μP card available. Forms the heart of our Z-1 and Z-2 systems. Also a direct replacement for Altair/IMSAI CPUs. Has 4-MHz clock rate and the power of the Z-80 μP chip. Kit (Model ZPU-K): $295. Assembled (Model ZPU-W): $395.

MEMORIES

- 16K RAM. The fastest available. Also has bank-select feature. Kit (Model 16KZ-K): $495. Assembled (Model 16KZ-W): $795.
- 4K RAM. Bank-select allows expansion to 8 banks of 64K bytes each. Kit (Model 4KZ-K): $195. Assembled (Model 4KZ-W): $295.
- THE BYTESAVER — an 8K capacity PROM card *with integral programmer*. Uses high-speed 2708 erasable PROMs. A must for all computers. Will load 8K BASIC into RAM in less than a second. Kit (Model BSK-0): $145. Assembled (Model BSW-0): $245.
- 16K CAPACITY PROM CARD. Capacity for up to 16K of high-speed 2708 erasable PROM. Kit (Model 16KPR-K): $145. Assembled (Model 16KPR-W): $245.

I/O INTERFACES

- FAST 7-CHANNEL DIGITAL-ANALOG I/O. Extremely useful board with 7 A/D channels and 7 D/A channels. Also one 8-bit parallel I/O channel. Kit (Model D + 7A-K): $145. Assembled (Model D + 7A-W): $245.
- TV DAZZLER. Color graphics interface. Lets you use color TV as full-color graphics terminal. Kit (Model CGI-K): $215. Assembled (Model CGI-W): $350.
- DIGITAL INTERFACE (OUR NEW TU-ART). Interfaces with teletype, CRT terminals, line printers, etc. Has not one but *two serial I/O ports* and *two 8-bit parallel I/O ports* as well as 10 on-board interval timers. Kit (Model TRT-K): $195. Assembled (Model TRT-W): $295.
- JOYSTICK. A console that lets you input physical position data with above Model D + 7 A/D card. For games, process control, etc. Contains speaker for sound effects. Kit (Model JS-1-K): $65. Assembled (Model JS-1-W): $95.

PROFESSIONAL QUALITY

You get first-class quality with Cromemco.

Here are actual quotes from articles by independent experts: "The Cromemco boards are absolutely beautiful" . . . "The BYTESAVER is tremendous" . . . "Construction of Cromemco I/O and joystick are outstanding" . . . "Cromemco peripherals ran with no trouble whatsoever."

Everyone agrees. Cromemco is tops.

STORES/MAIL

So count on Cromemco. Look into these Cromemco products at your store. Or order by mail from the factory.

We wish you pleasure and success with your computer.

Cromemco
incorporated
Specialists in computers and peripherals
2400 CHARLESTON RD., MOUNTAIN VIEW, CA 94043 • (415) 964-7400

CIRCLE 4

CONTENTS

SUBSCRIPTION POLICY

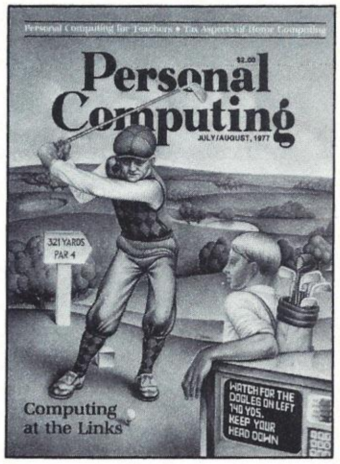
Subscription rates: Six issues – U.S., $8; Canada and Mexico add $2 for surface mail and $4 for airmail; all other countries add $4 for surface mail and $18 for airmail. Twelve issues – U.S., $14; outside U.S., postage rates doubled. Back issues (as available): $2 per copy in U.S.; $4 in Canada and Mexico; $5 in all other countries. Send subscription orders and changes of address to: Circulation, Personal Computing, 167 Corey Rd., Brookline, MA 02146.

Personal Computing is published bimonthly by Benwill Publishing Corp., 167 Corey Road, Brookline, MA 02146. Editorial office: 401 Louisiana S.E., Albuquerque, NM 87108; telephone (505) 255-8661 Publication office: 167 Corey Road, Brookline, MA 02146; telephone (617) 232-5470. Application to mail at controlled circulation rates is pending at Long Prairie, MN. Membership application in Audit Bureau of Circulation pending.

Personal Computing

JULY/AUGUST 1977 VOL. 1 NO. 4

COVER



Need help with your game? Artist Kim Behm putts the computer on the golf course. Cover design by Jane Higgins.

All editorial correspondence should be addressed to PERSONAL COMPUTING, 401 Louisiana S.E., Albuquerque, NM 87108.

We welcome submission of manuscripts for publication and pay competitive rates for material accepted. Authors should study the magazine for content and style to avoid sending inappropriate material at expense in time and postage.

Always include a stamped, self-addressed envelope if you want material returned. PERSONAL COMPUTING takes no responsibility for the safekeeping of materials not specifically solicited but attempts to respond promptly to submissions.

Letters

Dear Sir:

. . . The man who said he didn't want a kit went and bought one . . . from a computer store! The store: Doc's Computer Shop, 5755 Nolensville Road (Hickory Plaza Shopping Center), Nashville, Tennessee. The kit: an SWTPC 6800. The computer is up and running; however, I can't do anything with it because the TV typewriter (for input/output) and the cassette interface (for program storage) haven't been shipped yet. I can't blame SWTPC for non-delivery of the TV typewriter, because when Doc told me about SWTPC's new version with 64 characters per line and upper/lower case, I asked him to cancel my order for the old TV Typewriter II and reorder the new terminal. SWTPC says they'll ship the first units at the end of March. The new product announcement will be in the May issues of computer and electronics magazines. And since I can't use the cassette interface until I get the terminal, I'm not too unhappy with the relatively slow shipment. The demand seems to be so great that SWTPC is having difficulty with getting parts from its suppliers. I know, in my case, SWTPC made some parts substitutions with pen and ink corrections, showing the changes in the instructions and, in one case, substituted a 25V capacitor for the originally specified 16V. That, of course, caused no problems electronically, but the capacitor had to be mounted "side saddle" to clear a pc board in the adjacent slot. That's all the problems I've had with SWTPC, all trivial. The kit assembly went almost 100% trouble free. I caused myself a problem by applying power to the mother board with an 8835 installed backward (silly, silly), but as I'd used sockets, it was no trouble to replace the IC I'd ruined. A $2.00 mistake.

Without a terminal, I hauled my box down to Doc's one Wednesday evening to borrow his terminal (a Lear-Siegler ADM-3) for checkout. Apart from some operator error, the checkout went smoothly. We started by using the MIKBUG memory examine/modify function to write into the 6810 RAM on the MPU board. Nothing seemed to be too disastrously wrong until we got to location hex A00C. After writing 44C at that address, MIKBUG did not bring A00D up for examination, but 4400 instead. Strange. That caused some head scratching. Naturally you can guess what was happening. It seems MIKBUG uses A00C to store the high byte of the next address to be examined, and when we stored 44 into A00C, it munched the A0 that was (and should have been) there. I visited the store again Saturday when I checked out my memory board, using the diagnostics supplied by SWTPC. Keying in the hexadecimal code on the terminal was tedious (although not as tedious, I'm sure, as using binary switches on a front panel) and occasionally my typing reflexes would take hold and I'd type an "L" instead of a "1." My labor was well rewarded by rows of plusses and number signs, signifying that my memory was alive and kicking. Now all I need is my terminal so I can run Tiny BASIC, Tic-Tac-Toe and other goodies.

Bud Hamblen
Nashville, Tennessee

See Reader Hamblen's letter in the March/April '77 PERSONAL COMPUTING. He puts us all to shame in his diligence and patience. All for Tic-tàc-toe?

Dear P.C.,

After reading your magazine, I don't feel bad that my husband's buying a computer instead of my new electric stove I was supposed to have. I'm going to program it to do my grocery shopping. Thanks for a readable computer magazine.

Mary Capuccilli
Toledo, Ohio

Dear Editor,

I've really enjoyed your first two issues, especially Henry Gilroy's article on becoming a computer hobbyist in the first issue and Robert Rossum's article on "Waldo" in the second issue. I realize the latter article was written very futuristically, but given the right circumstances, it might happen sooner than expected. If a backer could raise enough to cover the cost of equipment to the United States Robotics Society and the event were held in Texas Stadium instead of the Astrodome, it could happen quite soon. I mentioned Texas Stadium (home of the Dallas Cowboys in Irving, Texas) rather than the Astrodome, because in March of each year it's converted into a motocross track which would make an ideal field for "Waldo" That would cut the initial investment by a considerable sum and if the Wide World of Sports would do a special on the Wide World of Technology, it could be a financially sound undertaking next year.

Danny S. Ferguson
Arlington, Texas

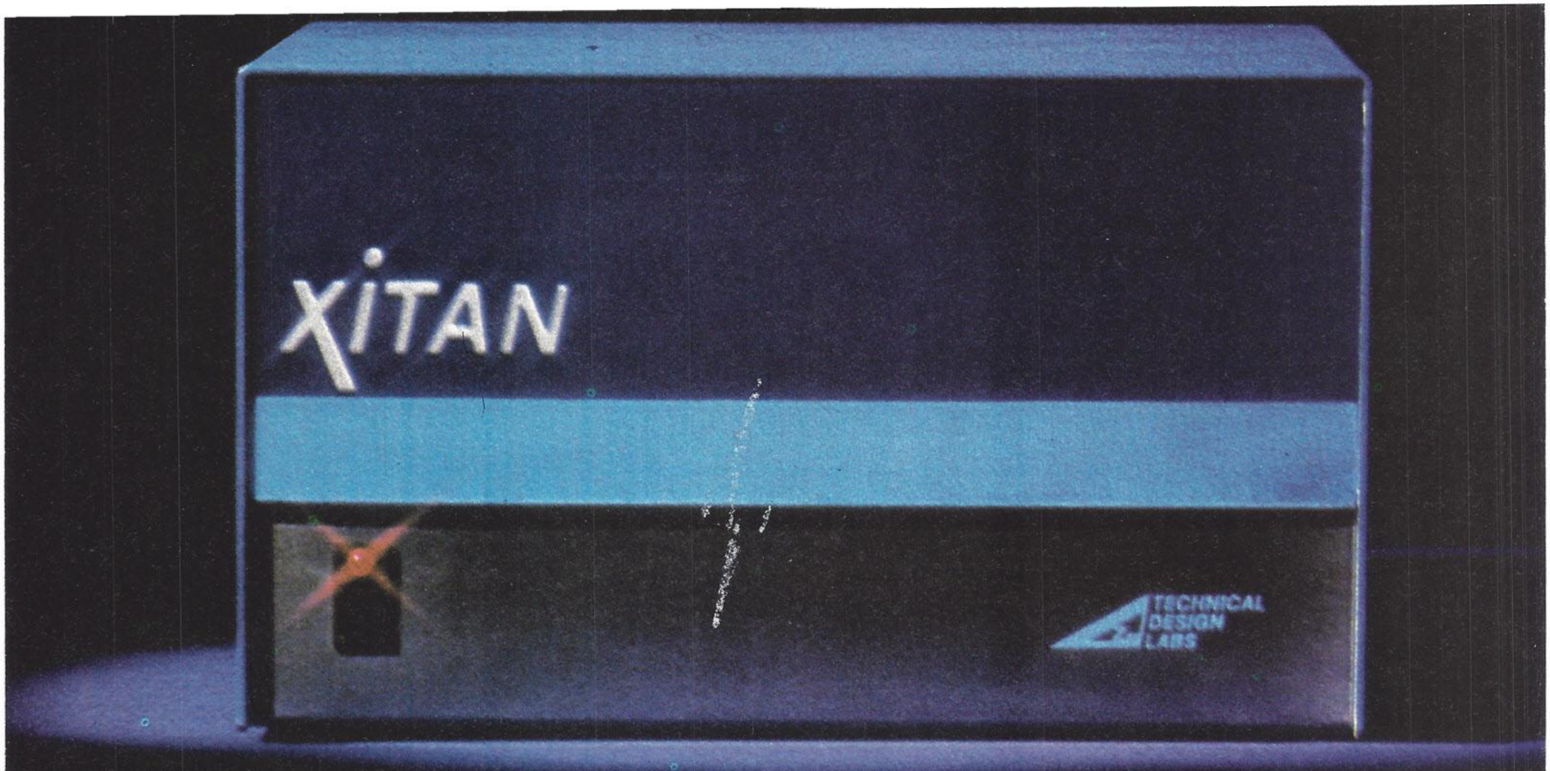
Actually, money is not the greatest obstacle to development of WALDO competitions; the more difficult problem is to find time and a way to let clever, independent amateurs develop their private ideas. All proposals are welcome. We are impressed by Reader Ferguson's a priori conviction that the Astrodome isn't necessarily the ideal place for WALDO, but that some other place in Texas might be. Go, you Longhorns!

Dear Editor,

. . . You might let your many readers know that, contrary to the article by Jeff Raskin (Big Computer, Little Computer – March/April, Vol 1, No 2, Page 31-35) a computer with 65, 636 memory locations (bytes) is not a 65K computer as stated on page 34. It is a 64K computer. 1K = 1,024; 1,024 x 64 = 65,636. No matter how large or small the computer, 1K is always 1,024 (either bits or bytes).

If this is the worst error I ever find in your magazine, you have earned high acclaim.

Keith L. Alloway
Seattle, Washington



IS YOUR DOLLAR BUYING AS MUCH POWER & FLEXIBILITY AS TDL'S XITAN SYSTEMS PROVIDE?

OUR CUSTOMERS SAY THINGS LIKE THIS ABOUT TDL PRODUCTS:

"... the best CPU board I've put together ... excellent parts ... worked right off without trouble shooting." .... JRG
Cambridge, Mass.

"Great product." .... KMM
Bella Vista, Ark.

"... high quality components, good engineering & complete documentation ... up and running without any problems." .... WP
Seattle, Wash.

"Excellent." .... Col. DWW
Santa Maria, Calif.

"Very impressed with superb quality." .... SK-L
Boston, Mass.

XITAN *alpha 1*

When we combined our highly praised ZPU board and our System Monitor Board, we defined the standard for the industry; we integrated more power and flexibility in two slots of our motherboard than most other systems can muster using five or more boards. When we put this setup into our rugged aluminum case we created the first XITAN system, the *alpha 1*. By adding a CRT terminal and/or teleprinter you will have a complete computer system.

KIT: $769 ASSEMBLED & TESTED: $1039

XITAN *alpha 2*

By adding a Z16 memory module and our PACKAGE A software to the *alpha 1* we created a second XITAN system, the *alpha 2*. Thus, a complete and extremely powerful micro-computer system emerges well worthy of you who are operating at the most sophisticated levels. The XITAN *alpha 2* provides you with 18K of RAM, 2K of ROM, 2 serial I/O ports, 1 parallel I/O port, our 1200 baud audio cassette interface as well as our extraordinarily powerful software package which includes 8K Basic, the Text Output Processor, the Zapple Text Editor and the Relocating Macro-Assembler. Add your own I/O device and GO ... with the most powerful and flexible micro-computer package ever offered.

KIT: $1369 ASSEMBLED & TESTED: $1749

**IF YOU ARE A BEGINNER, YOU WON'T EASILY OUTGROW THE XITAN SYSTEM.
IF YOU ARE AN ADVANCED USER, YOU WILL DISCOVER XITAN IS EXACTLY WHAT YOU NEED.**

†Write for descriptive brochure on the XITAN ***alpha*** series and system software. When you ask at your dealer, say "ZY-TAN."

ORDERING INFORMATION: Send check, money order or BankAmericard, Master Charge current number and expiration date. Shipping is usually made via UPS or UPS Blue Label. Specify other arrangements if you wish. Prepaid orders are shipped postpaid.

TECHNICAL DESIGN LABS
Z80

RESEARCH PARK BLDG. H 1101 STATE ROAD
PRINCETON, NEW JERSEY 08540 (609) 921-0321

Letters

Re: Jeff Raskin's Big Computer, Little Computer.

A nice story, good for non-technical people. Mr. Raskin was probably a good English major in school. He learned to write pretty well, with a sense of humor, but he must have flunked the technical courses. For his information K as in 65K etc stands for KILO as in kilogram, kilometer. It is exactly 1000, & he'd better learn it before we go metric. How could you as editors let something like that get through? The field is confused enough without confusing people more.

John Dodds
San Francisco, California

We're in the crossfire here and it's our own fault. The fact is that "K" is ordinarily used in this field to represent 2^{10}, as Reader Alloway observes. It's easier to say "4K" than "four-thousand-and-ninety-six." On the other hand, Reader Dodds is correct in assuming that "K" stands for 1000 and the term is commonly used among technical folk in odd combinations. "That caper cost the company 5K," for which read "five kilobucks." In richer circles, one may refer to megabucks. The safe course is to assume that "K" is an approximation of one thousand and to examine the context carefully to figure out details.

Dear Sir:

I'd like to ask a bit of help from PERSONAL COMPUTING for a problem: as I said earlier, I am a teacher in a business school where some students are quite a lot interested in computers and personal computing, even if personal computing is still almost unheard of in France. (It should not be for long, because one of our students intends to open a computer shop before September!) These HEC students are mainly BASIC users, but have also some notions of machine language, compilers and interpreters, and would like to work for 2 to 3 months in firms engaged in personal computing, be it on the hardware or the software side. Their English lacks some practice, but 2 weeks in the States should be a sure remedy to this. So, if you know about opportunities, please let me know. Or maybe you can give some audience to my problem, forwarding towards myself and HEC (address stated below) any answer or inquiry. (HEC is one of the most famous French business schools and I think that in fact I can find people even for non computer oriented applications). A difficulty is maybe that such applications are intended for this summer, and that I am not sure you will get this letter in time to make something; let's try however. I think that such contacts can be interesting for everybody, even if they materialize only next year.

Waiting for some help from PERSONAL COMPUTING,

Bernard Savonet
Departement Informatique.
C.E.S.A.
78350 Jouy en Josas
France

Sirs:

I enjoyed your brief article "Bearing in on Backgammon" in the March/April issue of PERSONAL COMPUTING. Your tactful proposal for a backgammon tournament in connection with the 1977 NCC is fair enough. Although time is short, I am sure that the the local steering committee, headed by Dr. Isaacson, would listen to your proposal. If the time is too short, why not consider NCC '78? It will be held in Anaheim, California, but run by a San Francisco Bay Area resident committee, under the direction of Mr. Steven Miller of the Stanford Research Institute, Menlo Park, California. It just happens that George Glaser also lives in the same area, so he might be persuaded to be one of the organizers working in connection with Steve's committee. In any event, please send me a copy of George's questionnaire, and let me know if I can help.

Robert W. Rector
Executive Director
American Federation of Information Processing Society, Inc.

It was very much too late for NCC in 1977, but '78 is still ahead of us and backgammon enthusiasts may wish to join this activity. If you'll write to the PERSONAL COMPUTING editorial office, we'll forward the material to any committee that forms.

Rich Man, Poor Man, Merchant, Physician, Teacher, Lawyer, Student, Musician....

There's an iCOM Floppy Disk System for Everyone!

More Uses
People from every walk of life are adding iCOM® Floppy Disks to their microcomputers for such diverse tasks as payroll, inventory control, mailing lists, game playing, record keeping, parts ordering . . .

We've uncovered some innovative applications, too: The sailboat architect who puts equations and algorithms on an iCOM disk to test his nautical theories; the student who has automated a bowling alley; the iCOM dealer who designed an environmental control system for a university.

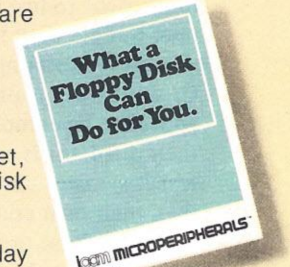
More Speed
These users have found iCOM floppies to be much faster and more versatile than cassette or paper tape. With iCOM, programs can be loaded in seconds; files updated in minutes; hundreds of programs can be stored on a single disk.

More Models
iCOM has Frugal Floppies™, Dual Floppies, Microfloppies™ (using the new 5¼" diskette), and other new approaches to floppy disk systems. Each is hardware and software compatible with Altair™, IMSAI, Poly 88, Sol-20 and other microcomputers using the Altair S-100 bus format.

More Software
Then there's iCOM's famous software: Powerful field-proven FDOS-II with macro-assembler, string-oriented text editor, and file manager. Plus easy-to-use compatible 8K Disk BASIC. Each with super features such as: named variable length files, auto-file create, open and close, multiple merge and delete . . . and more.

More Backup
We've been building floppies for microcomputers for more than 3 years. Long before the rest. Thousands of systems are operating perfectly in the field. And we're part of Pertec Computer Corporation, one of the largest manufacturers of peripherals, microsystems, data entry products and data processing systems. We'll be around whenever you need us.

More Dealers
Maybe not in quantity, but in quality. We've chosen our dealer network carefully to assure you of assistance every step of the way. Our prices are right. Our delivery is fast. Our dealers are experienced and knowledgeable.



Must Reading
Our free booklet, "What a Floppy Disk Can Do for You" is must reading. Send for yours today or visit your dealer.

icom® MICROPERIPHERALS®

a division of Pertec Computer Corporation

6741 Variel Ave., Canoga Park, CA 91303 U.S.A.
Tel. (213) 348-1391 TWX 910-494-2788

Letters

Dear Sir:

The most widely available STAR-TREK kind of game is the one in which a single ENTERPRISE has to rid the galaxy of Klingon battle cruisers.

In real life, if Kirk ever had to face more than one enemy vessel, he wouldn't even attempt to win in combat. He would tell a lie and then say "OK, Scotty, warp nine! Let's get this ship out of here!" Aye, Captain.

Therefore, to give the Enterprise a chance against so many, the Klingon ships are almost completely impotent. They can do no lasting damage to a battle ready starship. A Klingon will continue his little attack until he runs out of energy and effectively destroys himself.

This suggests a strategy in which the Enterprise enters a quadrant in which a Klingon is hiding, waits for the Klingon to wear himself out, and then retreats to a Starbase for repairs before seeking out the next Klingon. This strategy is not often used because it is boring (and non-violent) and because the player does not know where the Starbase is. How often did Kirk, in calm space, find himself unable to locate the nearest Starbase?

Why are the Klingons to be destroyed? Because, according to the scenario, the galaxy has been invaded by the dreaded Klingon menace. But what evidence do you see of a menace? The galaxy is full of Klingons, it's true. They have as much right to live in this galaxy as anyone. But are they threatening any planets? Are they mustering, preparing for war? Or wandering as pirates? No. They appear to be completely peaceful until their quadrant is entered by a warship which is many times more powerful than their own cruisers, commanded by a rookie whose orders are to search and destroy and not to ask questions or negotiate peace. Would you condemn them for defending themselves under these circumstances?

Now, what kind of starship captain would blindly follow orders to totally destroy a peaceful and reasonable people? We are talking about racism and genocide on a galactic scale. The winner of this game has committed the greatest and most hideous atrocity known in all the written histories of all the known peoples of the Universe.

In my own estimation, this is a grave misuse of the computer.

Now, don't get me wrong. Star Trek was good television, and in spite of its many flaws, I loved it dearly (except for the episode on the planet Scallos), and it is a good source of computer games. Parts of it are worth imitating. Parts, such as sadism and militarism, are not.

When the Enterprise was sent on its three year mission to seek out new worlds and civilizations, and to go where no man had gone before, was it to gain knowledge and understanding about the Universe to bring us, as a race, closer to fulfilling our destiny, or was it to find new markets and new sources of raw materials and territorial expansion for the Federation?

When you design the next generation of Star Trek computer games, keep in mind that Captain Kirk received advice from a Science Officer and a Surgeon.

Douglas Crockford
Newport Beach, California

Reader Crockford's earnestness called for publication of this thoughtful letter in its entirety. The Startrek phenomenon is amazing in the grip it has on many viewers and players. "In real life, if Kirk . . . " Real life? REAL LIFE?

PERSONAL COMPUTING has been offered some well-thought-out alternatives to Star Trek. The fact is that the game itself has no great appeal; it's just that the TV images linger on with a real-life quality.

Dear P.C.,

This home computer business has gone too far! Four years ago, my wife took the money I was saving for a tractor mower and bought a voltmeter and an oscilloscope. Three years ago, I planned to get a golf cart, but she used the money for some solar heating panels. Last year, my 750cc motorcycle turned into a geothermal coupler. But this year was the worst! I had put away a lot of money for a scuba outfit. Then she started reading PERSONAL COMPUTING. Now she has a microcomputer, a graphics terminal, and a line printer, and I have no tractor mower, no golf cart, no motorcycle, no scuba outfit. Phooey on you.

Jim Sprier
Cambridge, Massachusetts

Now. THE PERFECT MICROCOMPUTER

VECTOR 1

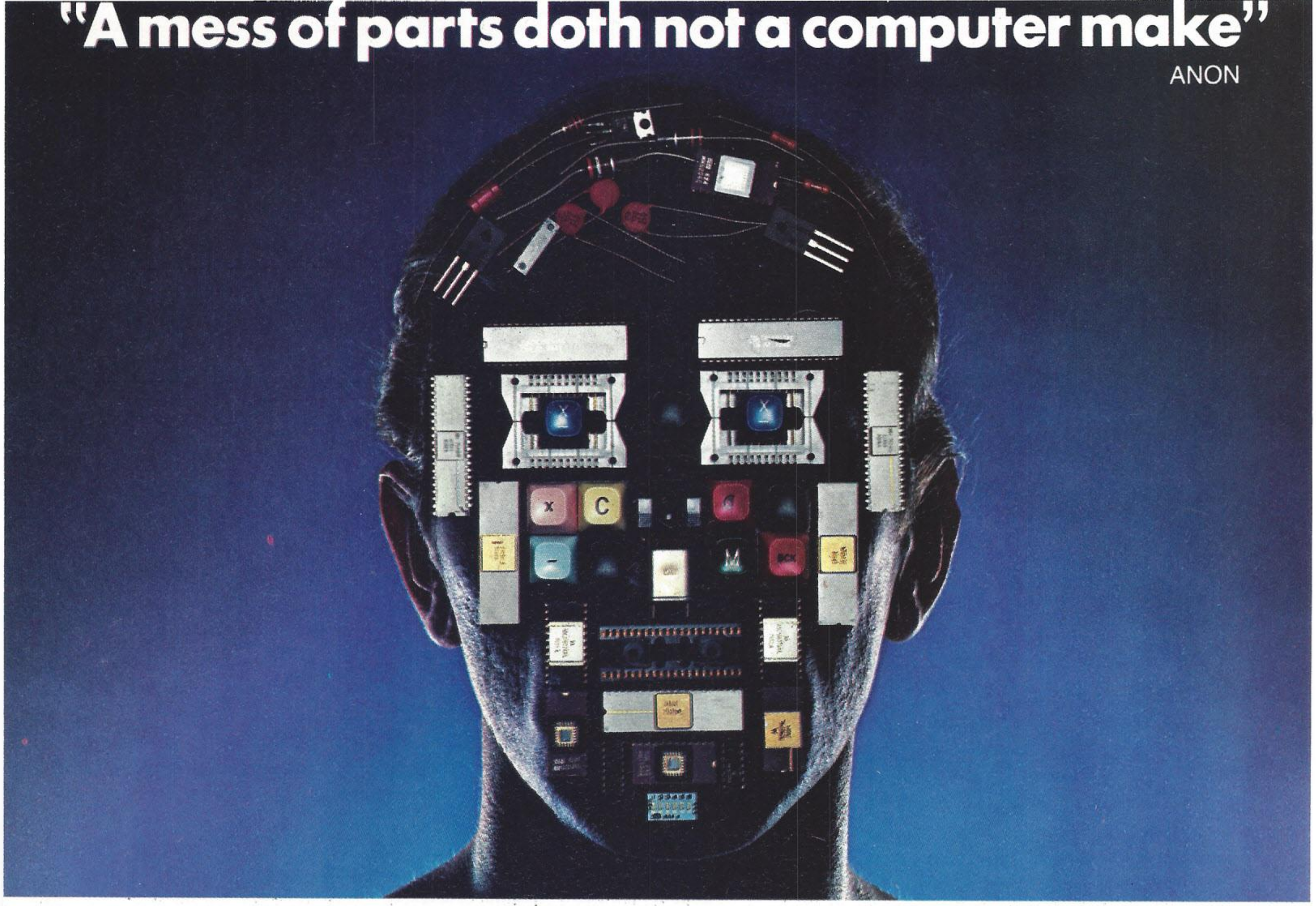
It's a two touch affair

VECTOR 1 is based on the 8080A Microprocessor and S-100 Bus -

EASY TO ASSEMBLE, EASY TO ACCESSORIZE, EASY TO BUY AND VERY EASY TO OPERATE.

VECTOR GRAPHIC inc.™

717 LAKEFIELD ROAD, SUITE F
WESTLAKE VILLAGE, CA 91361 • (805) 497-0733



"A mess of parts doth not a computer make"

ANON

Undoubtedly ANON was talking about some of today's PC computers. Exciting as they may be, most of them give you a few surprises you really don't need.

For example you have the board all wired, and suddenly it occurs to you, "Where's the TV interface?"

Surprise! (The TV interface is extra and costs $200 plus.)

"And the keyboard interface — where's that?"

Surprise! (Add another $185 extra.)

"How about an audio cassette interface so you can store and enter low cost programs and large amounts of data?"

Surprise! (Add $175 more.)

"And what about software?"

Surprise! (Each program will cost you $20 plus.)

See what we mean?

The new Sol PC Terminal Computer. It's a complete kit, hardware, software and no surprises for $475.

It's the first small computer conceived as part of a total system.

Not only do you get everything you need, but our systems approach is more economical, too. The $475 kit (fully assembled and tested, $745) includes all of the following as standard features:

Video display interface.
Keyboard interface.
Audio cassette interface.
8080 microprocessor.
512 eight bit bytes of PROM on a plug-in personality module.
2048 eight-bit words of RAM.
A 1024 character video display generator.
Serial and parallel interfaces for connection to external devices and an edge connector for memory expansion.

Optional extras include a power supply, video monitor, ASCII keyboard, walnut case, floppy disk system, high speed paper tape reader, PROM programmer, and color graphics interface. We also have a library of low cost software that's the best and most extensive in the industry.

In short, you get a complete, powerful, small computer and smart terminal that's fully expandable to a larger Sol system, as your requirements grow.

Surprises are great at Christmastime. But when you're opening the box of your new computer kit, you want everything you need to be there.

And with Sol, it is.

To order or to get more information write Processor Technology, 6200 Hollis St., Box L, Emeryville, CA 94608. (415) 652-8080.

Processor Technology Corporation

The Sol PC Terminal Computer

CIRCLE 8

MEMO FROM THE PUBLISHER

Shakeup in Personal Computing

The recent Computer Faire in San Francisco may have marked the high point of the personal computing market as insiders now know it. The shakeup that people have been talking about is either already here or just around the corner.

There were 13,000 people at the Faire and for the most part they are enthusiastic and eager to buy products. They are intelligent people who want to do something with a personal computer. Unfortunately, except for the programmers and engineers in the crowd, most of them aren't willing to dig inside their machines, build kits and fix the many problems you find with today's microcomputer. People don't fool with the insides of their TV's and stereos and they won't be any different about their computers.

In our March/April issue, Jim Blackman, an independent marketing consultant.for a number of large companies, called personal computing, "America's largest cottage industry" and how marvelously true that was in San Francisco. There was aisle upon aisle of curtained booths, for the most part filled with wholesale electronic parts companies, one man book publishers, garage operation manufacturers with names like "Cheap, Inc.", recently created software houses, and T-shirt vendors. Intermixed with these businesses were an emerging group of professional looking companies, a handful of IC chip makers like Intel and a few suprises like 3-M Company.

The emerging professionals include companies such as Digital Group, IMSAI, Mits, Apple, Processor Tech . . . etc. Most were one or two man companies not too long ago and each has its own particular aims and problems. As the mainstay of a 30 million dollar market, they are an impressive group.

And they'll have to be just that, if one is to take seriously the May 16 Industrial issue of BUSINESS WEEK. In a six page special report titled "The Coming Boom in Home Computers", BW reports that companies entering the market include Tandy (Radio Shack), Heath, Data General, Commodore, Texas Instrument and possibly even Timex. Competing against these companies will take high volume and massive distribution.

It's Not All That Bleak

I'm not a soothsayer and really haven't the foggiest notion about the outcome of all this. But, the handwriting is on the wall. Many wild-eyed optimists who believed *their* market for personal computing products is endless are going to wind up bankrupt. But a few of them are going to make it. They will become "real" companies.

Actually, it is not all this bleak. The need for constant innovations in application software, higher language software, and hardware will keep thousands of hobbyists, experimenters, "hardware hackers" and "computer freaks" busy. And they will need cheap, readily available supplies – enough to keep the flea market set in business if not in the limelight.

What About Video Games

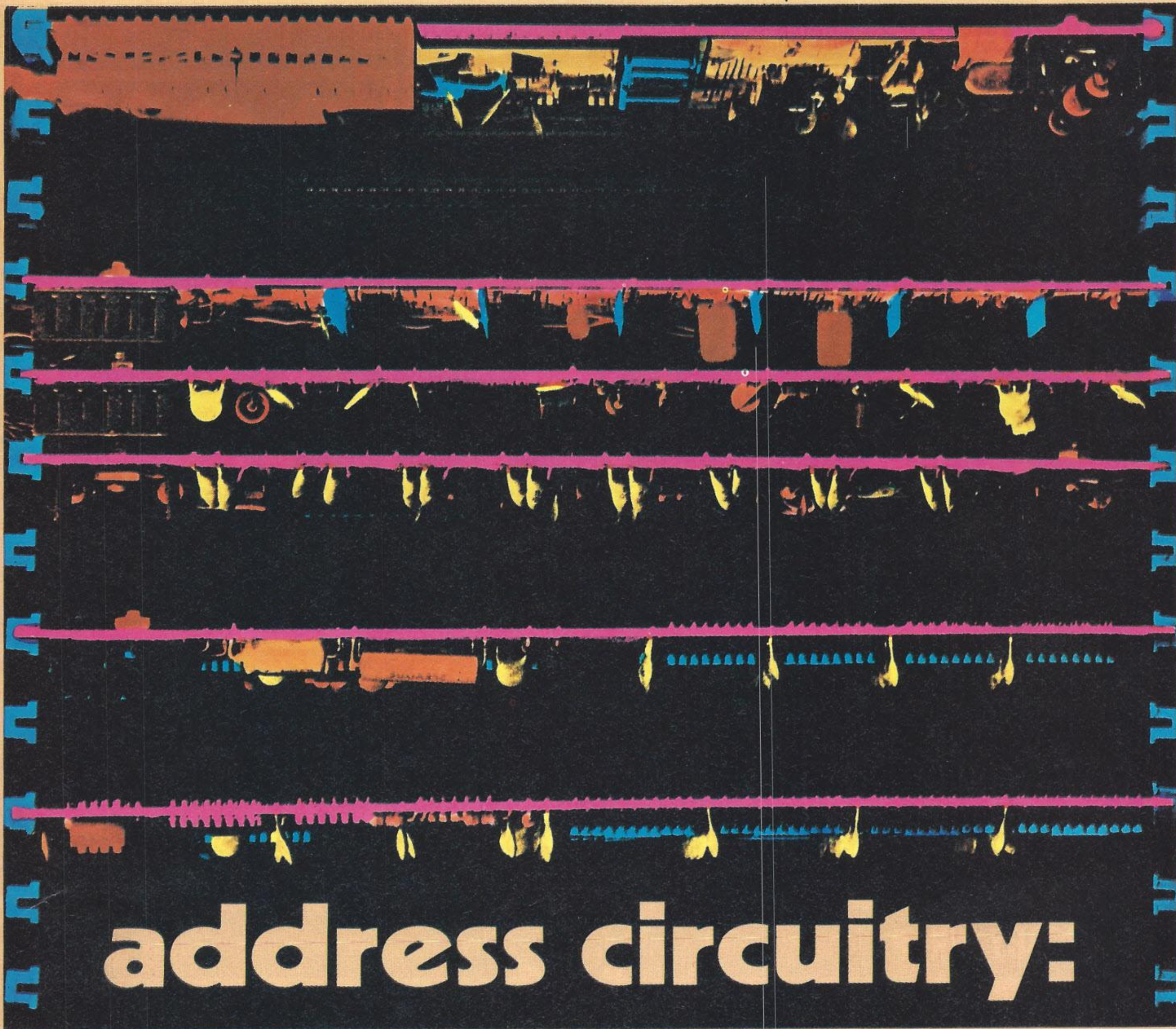
Recently, I've been playing with the Fairchild Home Entertainment Center, trying to determine for myself if there is a chance video games will turn into home computer systems.

In some ways the Fairchild system is a home computer. You can run an endless number of programs (mostly games) which are stored on plug-in ROM boards packaged to look very much like cassette tape cartridges. You can enter numbers by rotating the knob of the control stick. In Blackjack, you rotate it to the left to change the ten's place and to the right to change the one's place. In this way you can place bets between $1 and $99.

The Fairchild system was not intended to be a home computer and it is, of course, severely limited in the scope of its applications. However, it is a friendly way to introduce personal computer technology into the home and may help to lay the groundwork for a truly home computer center. And it's a real consumer product made by a "real" company with volume production. More on this in future issues.

David Bunnell

CIRCLE 9▶



Altair Computer Centers offer you direct access to the complete line of Altair microcomputer products. For demonstrations, information or service, visit the experts at your local Altair Computer Center today. They're located at these addresses:

ALTAIR COMPUTER CENTER
4941 East 29th St.
TUCSON, AZ 85711
(602)-748-7363

COMPUTER KITS (S.F. area)
1044 University Ave.
BERKELEY, CA 94710
(415)-845-5300

THE COMPUTER STORE
(Arrowhead Computer Co.)
820 Broadway
SANTA MONICA, CA 90401
(213)-451-0713

GATEWAY ELECTRONICS, INC.
OF COLORADO
2839 W. 44th Ave.
DENVER, CO 80211
(303)-458-5444

THE COMPUTER SYSTEMCENTER
3330 Piedmont Road
ATLANTA, GA 30305
(404)-231-1691

CHICAGO COMPUTER STORE
517 Talcott Rd.
PARK RIDGE, IL 60068
(312)-823-2388

THE COMPUTER STORE
OF ANN ARBOR
310 East Washington Street
ANN ARBOR, MI 48104
(313)-995-7616

COMPUTER STORE OF DETROIT
505-507 West 11 Mile St.
MADISON HEIGHTS, MI 48071
(313)-545-2225

THE COMPUTER ROOM
3938 Beau D'Rue Drive
EAGAN, MN 55122
(612)-452-2567

GATEWAY ELECTRONICS, INC.
8123-25 Page Blvd.
ST. LOUIS, MO 63130
(314)-427-6116

ALTAIR COMPUTER CENTER
5252 North Dixie Drive
DAYTON, OH 45414
(513)-274-1149

ALTAIR COMPUTER CENTER
110 The Annex
5345 East Forty First St.
TULSA, OK 74135
(918)-664-4564

ALTAIR COMPUTER CENTER
8105 SW Nimbus Ave.
BEAVERTON, OR 97005
(503)-644-2314

ALTAIR COMPUTER CENTER
611 N. 27th St. Suite 9
LINCOLN, NB 68503
(402)-474-2800

COMPUTER STORES
OF CAROLINA, INC.
1808 E. Independence Blvd.
CHARLOTTE, N.C. 28205
(704)-334-0242

COMPUTER SHACK
3120 San Mateo N.E.
ALBUQUERQUE, NM 87110
(505)-883-8282. 883-8283

THE COMPUTER STORE
269 Obsorne Rd.
ALBANY, NY 12211
(518)-459-6140

THE COMPUTER STORE
OF NEW YORK
55 West 39th St.
NEW YORK, NY 10018
(212)-221-1404

ALTAIR COMPUTER CENTER
3208 Beltline Road
Suite 206
DALLAS, TX 75234
(214)-241-4088 Metro-263-7638

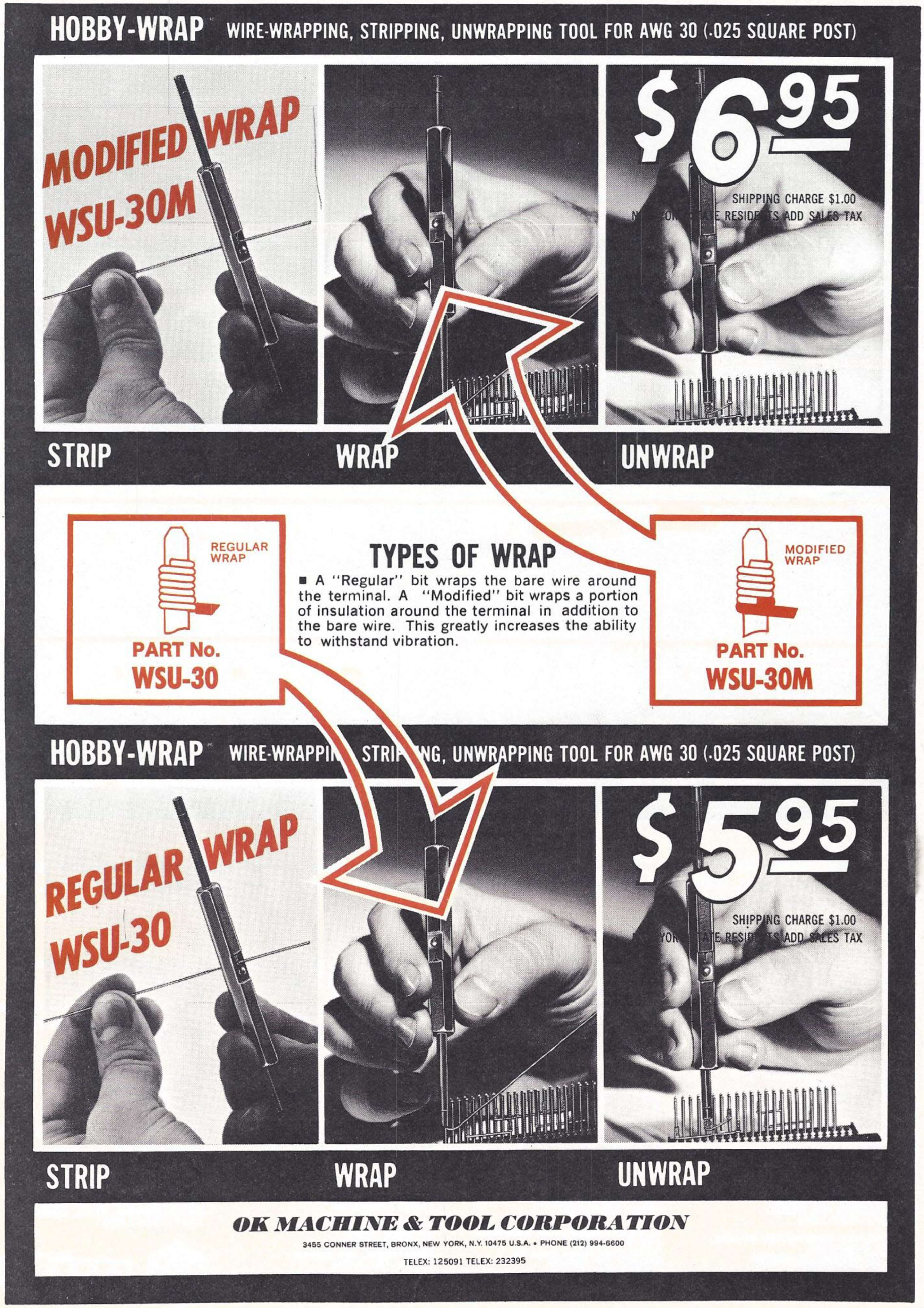
ALTAIR COMPUTER CENTER
5750 Bintliff Drive
HOUSTON, TX 77036
(713)-780-8981

COMPUTERS-TO-GO
4503 West Broad St.
RICHMOND, VA 23230
(804)-335-5773

MICROSYSTEMS (Washington, D.C.)
6605A Backlick Rd.
SPRINGFIELD, VA 22150
(703)-569-1110

THE COMPUTER STORE
Suite 5
Municipal Parking Building
CHARLESTON, W.VA. 25301
(304)-345-1360

mits



HOBBY-WRAP WIRE-WRAPPING, STRIPPING, UNWRAPPING TOOL FOR AWG 30 (.025 SQUARE POST)
MODIFIED WRAP
WSU-30M
$6.95
SHIPPING CHARGE $1.00
STRIP
WRAP
UNWRAP
REGULAR WRAP
PART No.
WSU-30
TYPES OF WRAP
■ A "Regular" bit wraps the bare wire around the terminal. A "Modified" bit wraps a portion of insulation around the terminal in addition to the bare wire. This greatly increases the ability to withstand vibration.
MODIFIED WRAP
PART No.
WSU-30M
REGULAR WRAP
WSU-30
$5.95
SHIPPING CHARGE $1.00
STRIP
WRAP
UNWRAP
OK MACHINE & TOOL CORPORATION
3455 CONNER STREET, BRONX, NEW YORK, N.Y. 10475 U.S.A. • PHONE (212) 994-6600
TELEX: 125091 TELEX: 232395

CIRCLE 10

A news release arrived recently

A news release arrived recently that is quoted here in its entirety:

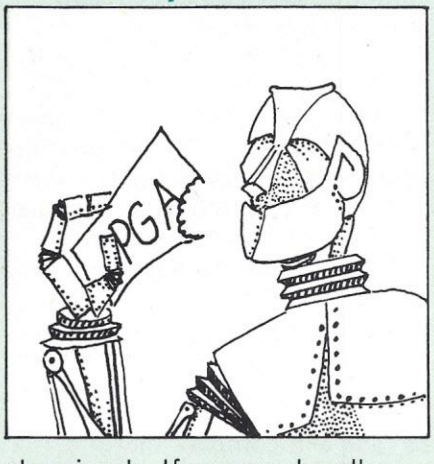
ZILOG Z80 MCB COMPATIBLE PROGRAMMABLE GAIN AMPLIFIER CARD (PGA)

The Signal Laboratories, Inc. Programmable Gain Amplifier card (PGA) is a Zilog Z80 MCB compatible accessory offering two channels of computer controlled amplification, each having both filtered and unfiltered outputs. Under control of the Z80 MCB the gain of each channel may vary from unity (Odb) through 54 db (70 db optional) in 2 db increments with gain accuracies of 0.1% (0.01% optional). Similarly the Z80 MCB may select any one of eight customer-predetermined bandwidths for each channel's filtered output. Unfiltered outputs are 3 db down at 50 KHz. The PGA is one of a family of Zilog Z80 MCB compatible accessory cards available from Signal Laboratories, Inc., 202 N. State College Blvd., Orange, CA 92668. For information call Del Flagg or Bill Chidester at (714) 634-1533. Price for quantities under ten is $395. Available from stock.

What do you suppose this is about? If you're a born-and-bred technical type, the meaning may be transparently obvious, but if you're a newcomer to computing, trying to figure out what this all means, Signal's release is not immediately helpful.

What would I do with one if I had it? Do people really buy more than ten at a time? Why? Surely this document conveys some information of value, but for whom? If I could decode this, would I be glad?

A phone call to Bill Chidester reveals that Signal isn't trying to conceal its meaning from the non-technician, and would be happy to explain the PGA card in English. "You see," says Bill, "computers are often used to interact with the real world, using inputs from measuring devices of various kinds as data on which the computer program operates. Usually the real-world information is in analog form, while the computer is a digital system. Analog-to-digital conversion is no problem, but when you are setting up a system, you want to have control of those an-



alog signals. If you can handle them automatically, it may save two days of adjusting by hand. That's what the PGA card is for, basically. Since it is designed to work with the Z-80 (no, not on an S-100 bus) it may be of interest to non-professional computer users."

If you plan to put your whole house under computer control — furnace, air conditioner, humidity controls, burglar and fire alarms — is this the sort of equipment you'd want to consider? "Yes."

If you're building a robot that interacts with the real world, would you find the PGA card interesting? "Possibly. There must be a lot of applications for this product and related items we make that haven't occurred to us in our industrial use of the technology. We sent the release to you as a sort of experiment, hoping that people from other small companies like ours might be looking in your pages for inexpensive computer systems that they can actually afford. We have nothing against the idea of explaining the virtues of these products in non-technical language. We just haven't done it. Maybe we should."

As many as twenty releases a day flow in to the Random Access editor. Perhaps a quarter of them are as crisply technical as this, tantalizing the non-insider, who has a hard time taking his mind off golf when he hears about the PGA.

Help over the wire

With 41,000 employees and an expanding $2 billion yearly business, Control Data Corporation has not yet shown much excitement about the little personal computing field. Even so, CDC is doing a number of fascinating things that may spill into personal computing as time passes.

For example, CDC operates a service called TECHNOTEC, designed to improve international communication in the marketing of new technology. Again and again, the wheel is reinvented, because the new inventor is unaware of previous work. TECHNOTEC is a huge pool of information accessible to anybody on the CDC worldwide computer network. A user with a small terminal somewhere can ask the system what's available in the

random access

way of round things associated with moving loads, and get a batch of information about wheels, who sells them, who has trade secrets on their use and applications information.

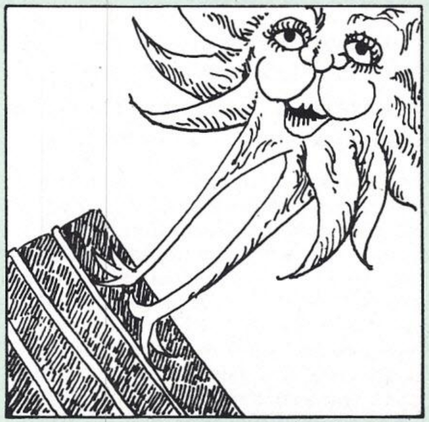
Presently, CDC is so serious about information treating the use of solar energy that the company is offering to finance use of TECHNOTEC to owners of solar technology who cannot afford the fee or are doubtful about the success of the exchange program. Perhaps this enthusiasm can have a significant influence on our energy situation. Personal computer users, looking forward to data networks, can learn from CDC and, perhaps, interact with their existing network on some basis. An inquiry to the TECHNOTEC network on computer networking technology may produce interesting responses.

CDC has also announced implementation of a legal service software package called PALLAS. Thinking of legal services, most of us expect great enthusiasm for using computers to search the files for legal precedents in statutory and case law. That's a worthwhile pursuit, though the task of entering legal data from time immemorial into the computer system is a big one, and some firms are offering service in that area.

Pallas, however, is devoted to handling the data, the documents, references and notes in specific legal cases. Suppose that you're an attorney dealing with a big case, say an antitrust suit that may go on for years and produce thousands of important documents that you must keep track of somehow. That's what Pallas is for, keeping lawyer's papers straight, not only in big lawsuits, but in handling the sea of paper involved in compliance with government regulations on equal employment, OSHA, and so on.

CDC is offering both Cybernet computer services and straight consulting by the Pallas team to companies with internal computer systems.

The value of such legal organizational systems was demonstrated a few years ago when two giant computer firms engaged in a huge lawsuit. The legal team of the plaintiff refused to mess with newfangled computer record-keeping, while the team of the defendant carefully built up a computer-based file through the long proceedings. The proceedings were **so** long and complex that the plaintiff began to lose track of documents, delaying matters still further. To the plaintiff's intense embarrassment, every time his attorneys searched for a lost paper, the defendant popped up with



the document, graciously offering it to the court, just to be helpful. Incidentally, the defendant won the famous case.

As personal computer users fight the battle against paperwork, CDC's example and service may both prove valuable.

And from the black chamber

Personal Computing will be running articles on cryptology — code making and code breaking — as it relates to personal computing in future issues. While most of us can enjoy the romance of the field, the entertaining stories of mystery,

espionage, dedication and betrayal, not everyone is interested and disciplined enough to get deeply into the practical aspects of crypt.

For the serious cryptologist, something new is on the scene, a quarterly journal, CRYPTOLOGIA, devoted to all aspects of cryptology, meeting what is felt to be a need for a public forum to exchange ideas related to cryptology. The journal will offer research papers, survey articles, personal accounts, reviews, educational notes and problems in mathematical, computational, literary, historical, political, military, mechanical and archaeological aspects of cryptology. The masthead of the magazine is even complete with funny pseudonyms. (Or, if we're mistaken, apologies to you, Cipher A. Deavours.) With luck, this journal will stimulate production of interesting articles for PERSONAL COMPUTING.

Write CRYPTOLOGIA, Albion College, Albion, Michigan 49224.

Ever more popular computing

For about five years now, Fred Gruenberger has been publishing a jolly monthly magazine called Popular Computing. Unblushingly, he demands $20.50 for a year's subscription to this 20-page, plain, mostly typewritten, no-advertising journal ($17.50 if you send the money with the order) and gets it from an enthusiastic readership scattered around the world. (Subscriptions are even more outside the U.S.)

Popular Computing is full of mathematical puzzles amenable

to computer or calculator solution, contests, commentaries and observations, and interesting discussions with correspondents. Useful information is also provided to those who can use it; e.g., π^{48} is 72978324682606403520 6280.5189276745690215459354 673957239829840096691073598 7607368.

This material is worthwhile chiefly because Fred Gruenberger is one of the real computer pioneers who got into the game when the computer was a pup and stayed in, working with everybody who is anybody in the field to move us along to our present state of technical advancement and amazement. Lately, he's been professing in the San Fernando Valley, teaching programming that's **programming!** And he's more than a bit crusty about his subject. "Kids come into my class all full of pride, because they've played with a microcomputer, written some trivial exercise program, and learned all there is to know. They are taking my class only because it's required to get a degree. Well, I try to teach them a few things about programming that they haven't already learned!"

For a sample of Popular Computing, send $2.50 to Box 272, Calabasas, California 91302. And never mind proposing trades for the publication, Dr. Gruenberger won't even trade subscriptions with PERSONAL COMPUTING. It's a seller's market and he's got the good goods.

Both tenor and bass

He could sing like Caruso
Both Tenor and Bass
And play on the Spanish
Guitar.

Thus is Ivan Skavinsky Skivar admiringly described in the old college song, Abdullah Bul Bul Emir. (If you don't know it, you're missing some elegant language.)

The song refers to a famous incident in which an operatic bass was rendered temporarily voiceless when he inhaled a fly or something, and Caruso, who was on stage with him, kept the show going by turning his back to the audience and singing the basso's aria while the man moved his mouth and recovered his composure. By all accounts, Caruso was truly an astonishing singer as

well as an interesting character.

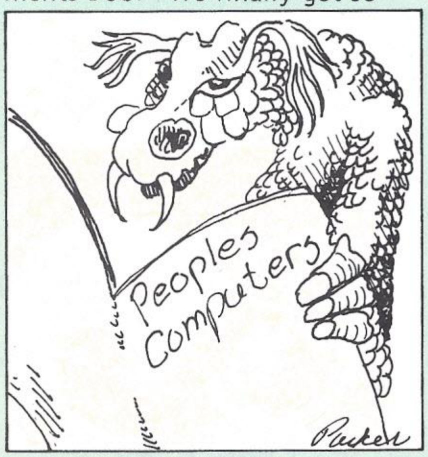
Unluckily, we can't hear the man sing, because the recordings he made, all before 1925, have built-in distortions and peculiarities inherent in the recording technology of the time. The recordings give us only a hint of Caruso's real voice quality.

Technology has advanced, and diligent digital engineers are finding computer processing techniques that take the hints available on the old records as a guide to reconstruction of the original sound. In fact, an RCA release called "CARUSO, a Legendary Performer," has been rebuilt this way with such success that the album became the best selling classical record in America within four months after its release.

random access

Pcc's reference book

PCC'S REFERENCE BOOK OF PERSONAL AND HOME COMPUTING is just being released as a "single source of basic, concise, detailed information in the field." With any luck (the book is not yet hot off the press, available for review), this will live up to its advance billing, because a good, comprehensive directory is sorely needed. Comments PCC: "We finally got so



tired and frustrated at searching for this information, that we decided to collect and publish it ourselves. Hundreds of sources for products and services, from the tiny one-man shop to the international corporation, are now available for easy reference."

According to plan, the manual will cross-reference stores, manufacturers, clubs, people, hardware, and software systems. The bare data will be supported by basic articles treating the fundamentals of the field so that users can develop an overview of the subjects that make the data more useful.

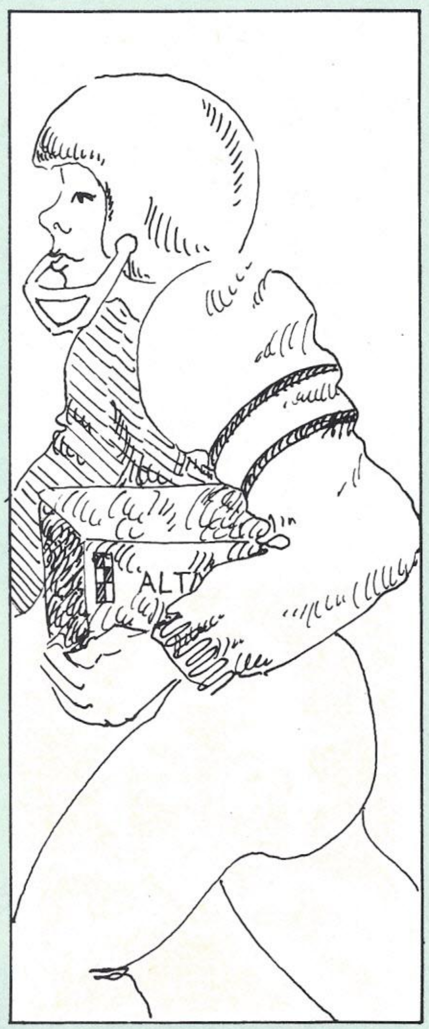
People's Computer Company (there's just something about that name that brings East Europe to mind) is in a uniquely strong position to produce such a work. PCC has been plugging away for several years as a force for good in the community under Dragon Bob Albrecht's influence, publishing a jolly bimonthly tabloid newspaper and the redoubtable Dr. Dobbs Journal of Computer Calisthenics and Orthodontia. (The tabloid is now changing its name to (sigh) People's Computers, and shifting to a nifty new format on different paper.) Most news in the personal computing world flows through PCC, and the organization is a gold mine of hard data, rumors, and good cheer.

If PCC's Reference Book of Personal and Home Computing lives up to its potential, it will be well worth its scheduled $4.95 (ppd) price.

Write PCC, Box E, Menlo Park, California 94025.

And they're off!

Taylor University is so pleased by the performance of its programming team in winning a Midwestern Regional BASIC programming competition last April, that the school is now agitating for more competitions, with a national championship meet to be held at Taylor in the spring of '78. Or weren't you aware of programming competitions?



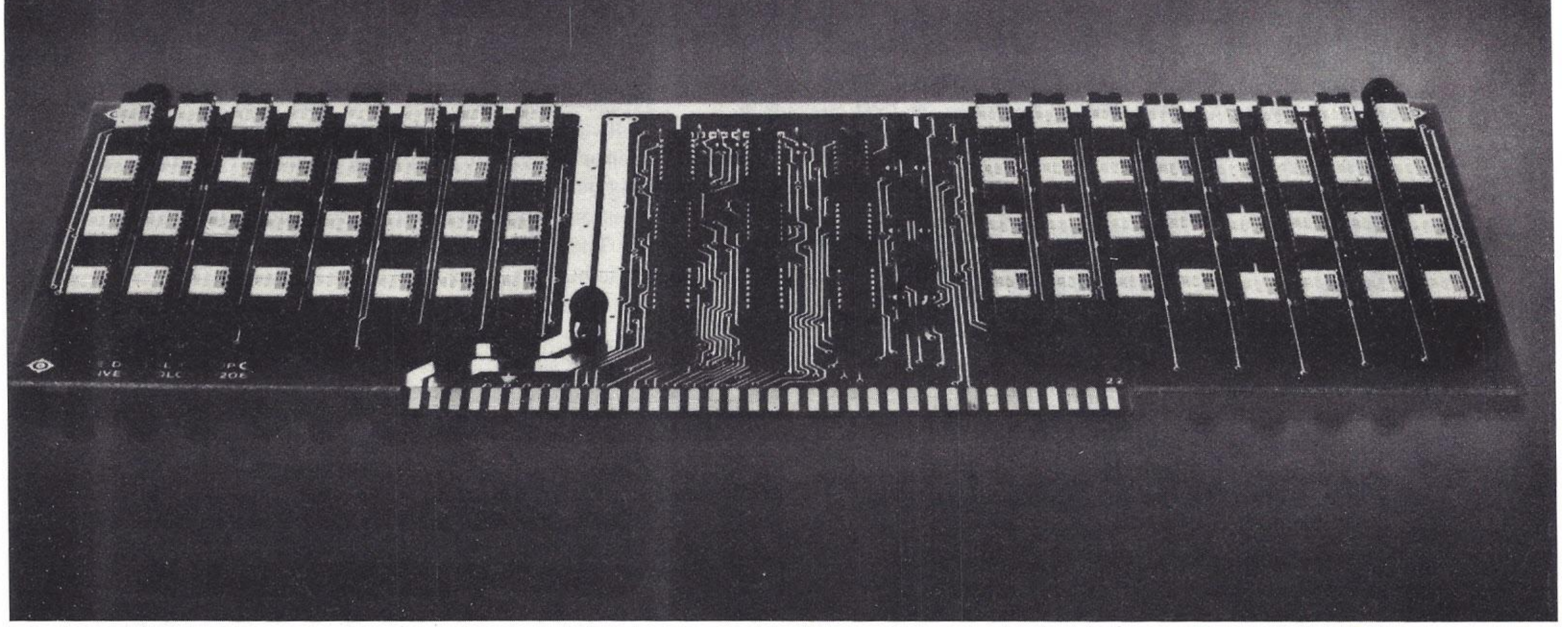
In the April event, some nine teams from as many schools were presented with five interesting problems for computer solution and allowed four hours to solve them. For example, they were to determine all of the ways in which a given football score could have been produced in a game under standard football rules. (What combinations of touchdowns, extra points and forfeits could generate a score of 24 to 17.) Or, given the day of the week of the first of a month in some year, can your computer print out a proper calendar for the year?

The only restriction in this competition was that the problems be solved with programs written in some acceptable dialect of BASIC. Any machine could be used, though, in fact, eight of the nine schools used the PDP 11/40 available at Taylor, while the ninth team phoned in its H-P 3000 at home. Presumably a team could have carried along its own Altair system, or whatever.

Random Access Part 2 begins on page 111.

CIRCLE 11 ▶

Memories are made of this.



32K. One Card. One low price. Only from the Digital Group.

Now, on only one *fully static* card, the Digital Group has squeezed in a whopping 32K of memory. Which, with a little quick addition, means a full 64K architecture now requires only 2 boards instead of 8. That's a 4-to-1 space reduction ...and leaves one extra memory slot on the Digital Group's standard motherboard still available for future products.

All this and one low price, too.
It just may be the best news of all. Our full static, assembled and tested 32K memory board is only $995. Now that's worth remembering. It's substantially less than our equivalent assembled 8K board prices. (Please note: We're initially offering this 32K board assembled only, but kit versions will soon be available, too—at even lower prices.)

Here's what you get.

Specifications:
- 32K on single card
- Speed—450ns. All of our current CPUs will operate at full-rated speed.
- Decoding—Lower or upper 32K bank
- Power—+5V only @ 4A
- Card size—12″ x 5″ (excluding connector fingers)

Features:
- May be intermixed on Digital Group systems with our 8K memory cards
- All data and address lines are buffered
- *Fully* static memories—EMM 4801 (450ns) or equivalent

Price:
32K board complete, assembled and tested $995.00

For all the memorable details, just fill out the coupon below. (Then all you have to remember is to mail it in.)

---

the digital group

P.O. Box 6528
Denver, Colorado 80206 (303) 777-7133

☐ **I promise to mail this in, so add me to your mailing list!**

Name ______________________________

Address ______________________________

City/State/Zip ______________________________

☐ Remember me? I'm already on your mailing list, but I need the memory spec sheet desperately.

With Colonel Colt's six-gun, the uneasy average man was equal to the dangers of the frontier. The uneasy average man at today's frontiers of knowledge can hold his own with the personal computer. It is:

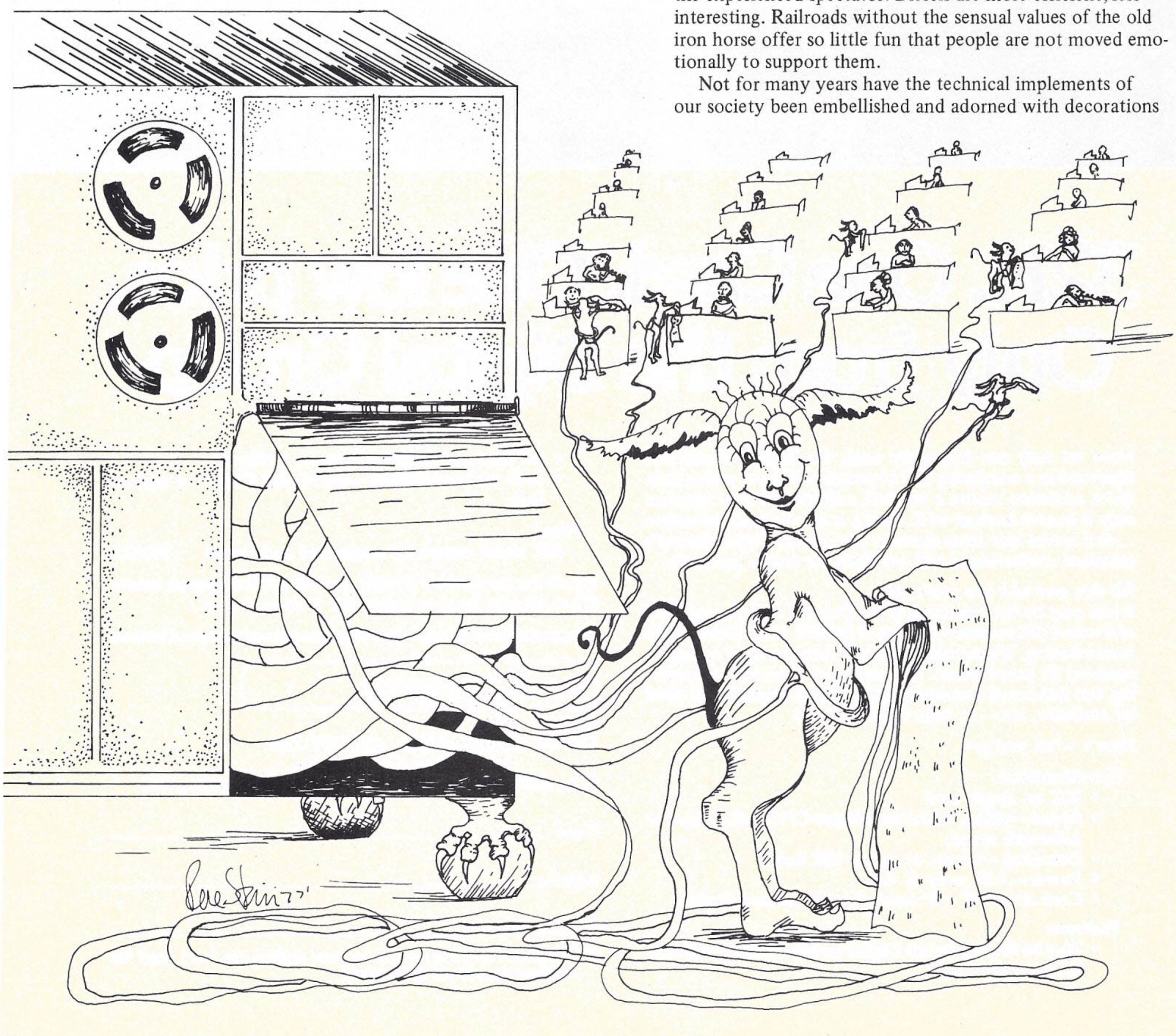
THE EQUALIZER

This may be the Golden Age of Technology, but in some respects it is less exciting and satisfying than the Brass Age of Technology. Perhaps we can do something about this, recognizing the problem.

One hundred years ago technology largely involved big mechanical systems that were fascinating to watch. Steamboats sported not only churning paddlewheels but great walking beams that rocked majestically with the action, huge pistons whose gleaming rods traveled yards at each stroke. The engines chuffed, puffed, roared and emitted dramatic bursts of steam.

The decline in the fortunes of our railroads may in part be owing to the abandonment of steam engines, those raging, snorting fire-and-iron demons whose passage thrilled even the experienced spectator. Diesels are more efficient; less interesting. Railroads without the sensual values of the old iron horse offer so little fun that people are not moved emotionally to support them.

Not for many years have the technical implements of our society been embellished and adorned with decorations



that pleased designer, manufacturer and customer alike. Does your stove at home boast an iron eagle on its oven door, cast in bold relief, marking the product as a proud spokesman for its maker? Probably not. Is the lathe in your shop supported by four straining steel satyrs with wicked half-smiles and glittering eyes, whose demoniac power is directed to holding the machine at its work?

Chances are that your bathtub is not supported on scary hawk's talons gripping iron orbs the size of tennis balls. You probably don't object when Sears offers to sell you a tractor that doesn't even have a brass steam whistle. There's no copper gingerbread on your Faraday cage, no brass bunting on your motorcycle. The most we expect in equipment decor is an occasional pinstripe or an expanded metal grille.

We can't hope to regain the fun and excitement of Victorian, brassbound, rivet-headed technology by returning to the hardware of yesteryear, but we may generate new fun in our technology if we try. Some of us suspect that the primary impetus to technical development is the fun of fooling with technology. Goodness knows, innovators are seldom rewarded with money and recognition; the road to Hell is paved with good inventions. It's the fun along the way that keeps us moving.

Computers have traditionally been such obscure, difficult-to-build devices that only a tiny percentage of our society has had fun with them and enjoyed the chance to innovate in the field. Their expense has been so great that the few buyers of computer systems have felt compelled to eliminate frills, stick purely to perceived function (discouraging innovation) and leave out the fun.

Outsiders are not much interested by conventional computers, drawn to them only long enough to take a quick look, yawn and go on. Computers themselves don't do anything overt. Their most dramatic activity is to blow hot air on your shoe.

Even computer-controlled peripheral equipment doesn't hold the attention of the passerby for long. The typewriters type. The plotters plot. Notice that moviemakers always use shots of tape reels turning, ignoring the mute, motionless boxes in which the computer itself resides. Personal computing is booming in spite of computers, surely, not because of their physical attractiveness. Non-professionals are now determined to get behind the blank, bland mask of computerdom to find out what's going on and influence it if they can.

The non-professionals may open the view so that other outsiders can see the busy works. There are no walking beams and pistons. The electronic action is so swift that it cannot be discerned, but there must be some way to convey a satisfactory sense of the action.

Can't we equip our personal computers with peripherals that *do* something of interest? Apart from decoration, which is not in fashion, we could develop signal systems, bells and whistles, sign language displays, that give us some sense of what the computer is up to in addition to providing bare output of formal data. Our perception of the "state of mind" of the system can be as useful, perhaps, as our perception of how an engine sounds.

We don't see human and animal thought processes directly, but we are deeply interested by the behavior that reflects the internal grinding of mental wheels.

We diagnose the performance of familiar systems like cars and refrigerators chiefly with the aid of subliminal cues, little hints and patterns that we can't describe consciously but of which we are profoundly aware. At the moment, few of us have any such sense of computer system operation. All our contact must be rational, studied, analytical. We have no feeling in our bones for what is proper and possible.

Might computers be more comprehensible if, instead of dinging a bell to tell us that a body of data has been printed out, they sent a little messenger to fetch us, a wide-eyed, smiling little critter on a long leash that bumbled about gently, carrying messages? Wouldn't the kids take to such a representative of the personal computer, eagerly learning techniques for training it to do tricks and perform chores, becoming acutely sensitive to variations in its performance?

Might the computer's degree of agitation, its level of activity, the percentage of its memory occupied, its backlog of scheduled operations be represented in pulsing color and patterns designed to mesh well with the human sensory system?

Most of us empathize with our automobiles as they struggle up steep grades. We speak words of encouragement as they strive to do the tasks we have set for them . . . and snarl when they fail at ordinary work. Isn't it inevitable that our personal computers will be thus personified, named, coddled, cursed and enjoyed as entities, not just dull, inanimate objects? We still admire The Little Engine That Could. Certainly Engineer Scott loves the Enterprise.

The professionals have had little time and no budget for developing logical processing systems with which an operator can enjoy a harmless, entertaining personal relationship of respect and affection. That's where much of the fun in technology is found. Personal computing may restore this fun to technology for the benefit of us all.

HOMEWORK

BY FRED AND JOAN SABERHAGEN

INSTITUTIONAL COMPUTERS ARE NOT NEW TO SCHOOLS, BUT PERSONAL COMPUTERS ARE NEW TO TEACHERS WHO MAY NOW PUT THESE REMARKABLE TOOLS TO WORK IN NOVEL WAYS WITHOUT A BUREAUCRATIC STRUGGLE. HERE A FAMOUS SCIENCE-FICTION WRITER AND HIS TEACHER/PROGRAMMER WIFE SPECULATE ON PERSONAL COMPUTER APPLICATION IN EDUCATION.

One argument often presented to school boards and taxpayers, with the intent of reconciling teachers' fulltime pay with their apparently short hours, is that a teacher's job does not end when the school bell rings. And indeed, if the job is to be properly done, it requires, on the average, another hour or two of effort per day after the kids go home.

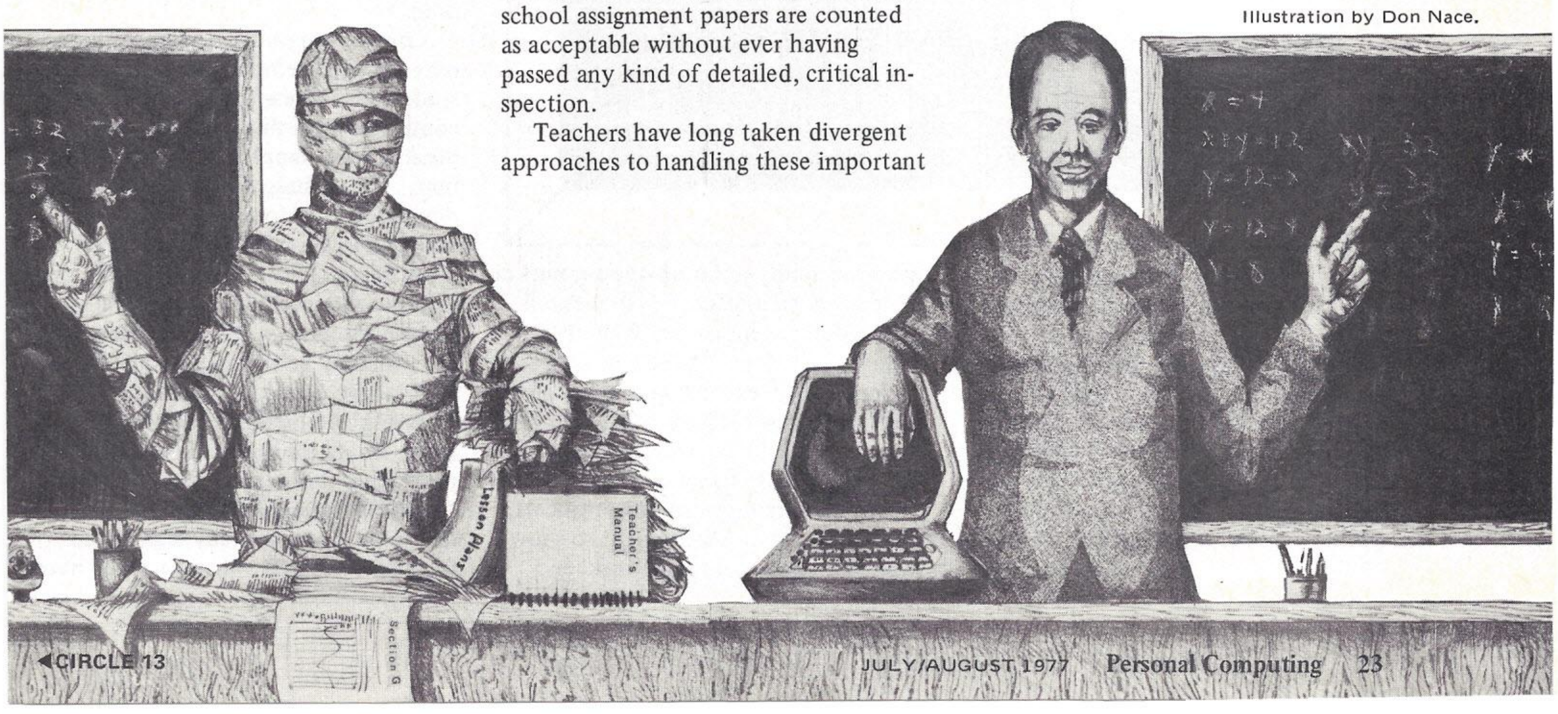
This extra labor on the part of the conscientious educator, performed either in the school building or at home, entails much shuffling of papers and rearranging of the information on them. An average teacher has perhaps five or six classes of twenty to thirty students each. On a normally busy day, almost every student will turn in at least a substantial portion of the day's classroom assignments or homework. Ideally, each symbol on each flake of paper in this storm will be checked for correctness as part of the teacher's homework, and each paper will be marked with a grade before it is handed back. These grades must also be recorded in the teacher's records. At intervals of a week or two or three, twenty to thirty test papers from each class are handed in to swell the load. Nowadays some – not many – teachers enjoy the services of paid full-time aides provided by the school, who can take over much of the burden of paperwork. Other teachers are lucky enough to have obliging and knowledgeable helpmates at home. Still, as with income tax returns, a great many school assignment papers are counted as acceptable without ever having passed any kind of detailed, critical inspection.

Teachers have long taken divergent approaches to handling these important off-stage tasks, even as they have to the more public portions of their jobs. Now the home computer can be employed to pull this load of mental donkey-work in any desired direction. It can also make some new approaches possible, and ought to be of considerable benefit to everyone involved.

Aid for the individual student

To begin with, happy are those students (or anyway, likely to learn something) whose instructor can harness the power of the computer to diagnose their specific individual difficulties with the subject matter. Happier still are they whose teacher can then use the same device to help plan more or less individualized

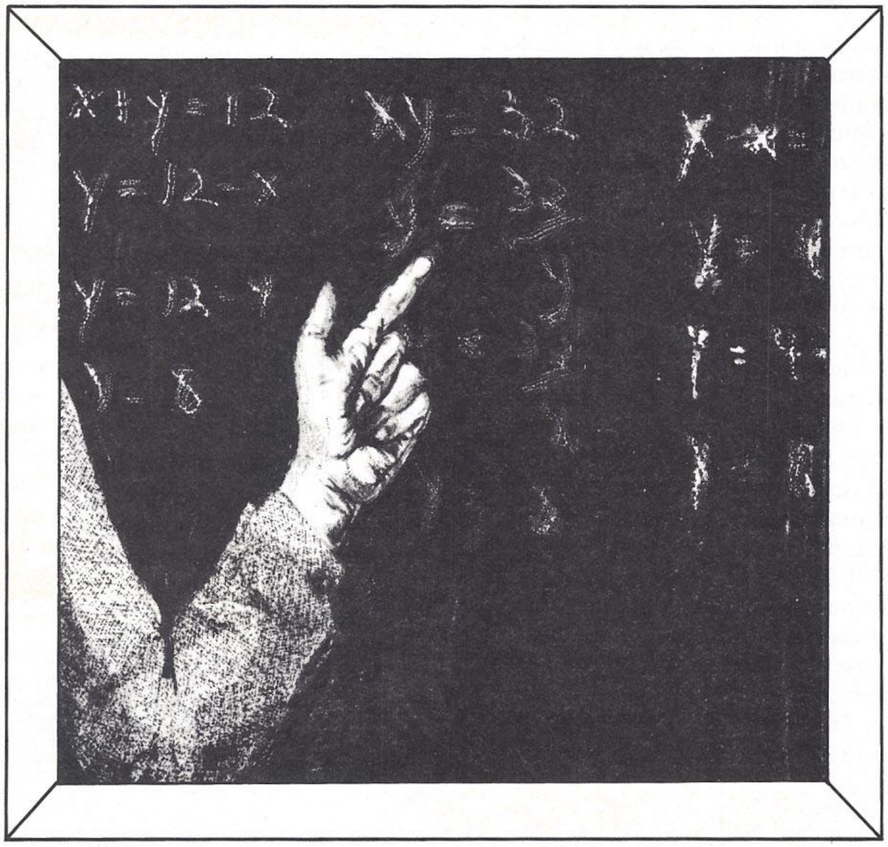
Illustration by Don Nace.



courses of study, concentrating neatly on the areas where the kids need help. These individualized study plans can be varied from week to week, or even from day to day, as the students master old difficulties and strike new snags. Of course sooner or later this kind of lesson planning is likely to be taken over by the school system's own computers. But in the budgets of many ordinary schools, teaching machines (and even hand-held calculators) seem years away – at least in numbers great enough to let them be counted among the schoolmaster's ordinary tools.

Although the learner in almost any subject can doubtless benefit from the teacher's intelligent use of a computer, there seems reason to believe that the math student may be the easiest one to help. Of all common school subjects, mathematics offers perhaps the neatest array of discrete facts. Answers to questions tend to be cleanly right or wrong. Also, the processes giving any particular pupil a hard time can be isolated accurately. (If Johnny misses Problems One and Three on the test, but manages to figure out Five and Six correctly, he seems to be doing all right at subtracting double-digit numbers until he runs into borrowing.) A personal computer program ought to be able to take note of just which kinds of problems a student cannot do, and recommend the proper pages of a textbook for remedial study. Fortunately, the same text is commonly used in a given school for five or six years running, so the time and effort spent in working it into the data base can be amortized over that period.

Of course this system of recommending study materials need not be limited to any one book. Let someone take the trouble to feed in enough information, and it can provide an instant bibliography of articles, books, and other materials relating to any of a large number of mathematical topics. Perhaps in this mode of use, the computer can be even more useful in other subjects. The materials can themselves be tested, by re-testing students who have studied them; materials shown to be ineffective can be dropped.



QUICK GRADING

For decades, enormous organizations such as the federal government have administered tests in multiple-choice form, using answer sheets capable of being scored directly by machine. Given this same capability at home, the amount of time saved by a teacher over a school year would be enormous. Even if answers must be transferred from paper to machine by eyeball and finger, great advantages result. The grade is computed automatically, and simultaneously recorded. (Presumably students will still be required to turn in worksheets, spot-checking of which ought to show who is actually doing the work, and who is copying from a co-operative classmate.)

Another substantial time saving accrues at the end of each marking period, when these recorded grades can be instantly compiled and quickly printed out. And, any student whose fate hangs in the balance at the eleventh hour can be easily furnished information on just what grade he must achieve on the final exam in order to pass the course.

At any point during the term a student's progress, or lack thereof, in particular areas of study, can be readily compared not only with the progress of his own classmates, but (if the data is available) with that of people in other classes or even those in other schools. This year's class can be compared with last year's, in minute detail.

AND STATISTICS

Today's public school teacher is normally confronted at least twice a year by what are known as the national averages. These numbers stand in the thoughts of professional thinkers on education much as batting averages figure in the minds of baseball coaches and sportswriters. Scholastic national averages are based upon the scores achieved by millions of students in biannual tests, administered nationwide by a few large testing companies. The tests are intended to measure basic skills in English, social science, mathematics, and science. Naturally any teacher whose students' scores on these tests go upwards faster than the norm ought to be beloved of the local administrators. And any teacher whose home computer contains the question-by-question scores of his or her students on the first nationwide test of the year – the one given early in the term – has constantly at hand a record of where each kid needs help.

Comparisons and correlations have only begun. The test scores of incoming students from one feeder school can be contrasted with those of kids from other schools supplying students to an upper school class. Given large enough samples, the performance of an otherwise unknown individual teacher, in some unseen feeder school, can be compared to the work of other men and women teaching at the same level, and can probably be shown by statistics to be above or below standard.

Books and methods can be similarly rated. Mr. Chips hopes to become a better teacher this year by using Textbook A and Educational Theory One. But did more people acutally learn more from him last year, when he relied upon

Book B and a large paddle? Theory One may work just beautifully for Ms. Splinters down the hall; administrators with home computers will perhaps be able to figure out just how far she and Chips should go in standardizing their approaches. (Administrators, by the way, also ought to be able to employ their home computers to advantage in the infernally complex task of scheduling next semester's classes, for a student body whose size and makeup is continually changing. Large school systems have already been using their own computers to do this for years.)

There would seem to be no end to the statistical feats practical. In this as in other applications of statistics, it will take some judgement to decide when the end of sense and useful meaning has been reached. Might tall girls with short family names learn trigonometry measurably faster than short boys who play basketball?

It is not new that social and legal repercussions should result from a teacher's decisions in the classroom. But now we can at least hope that those decisions, when backed up by the informed use of a computer, will tend to be better than in the past. The key word, as so often, is *informed.*

SELF DEFENSE

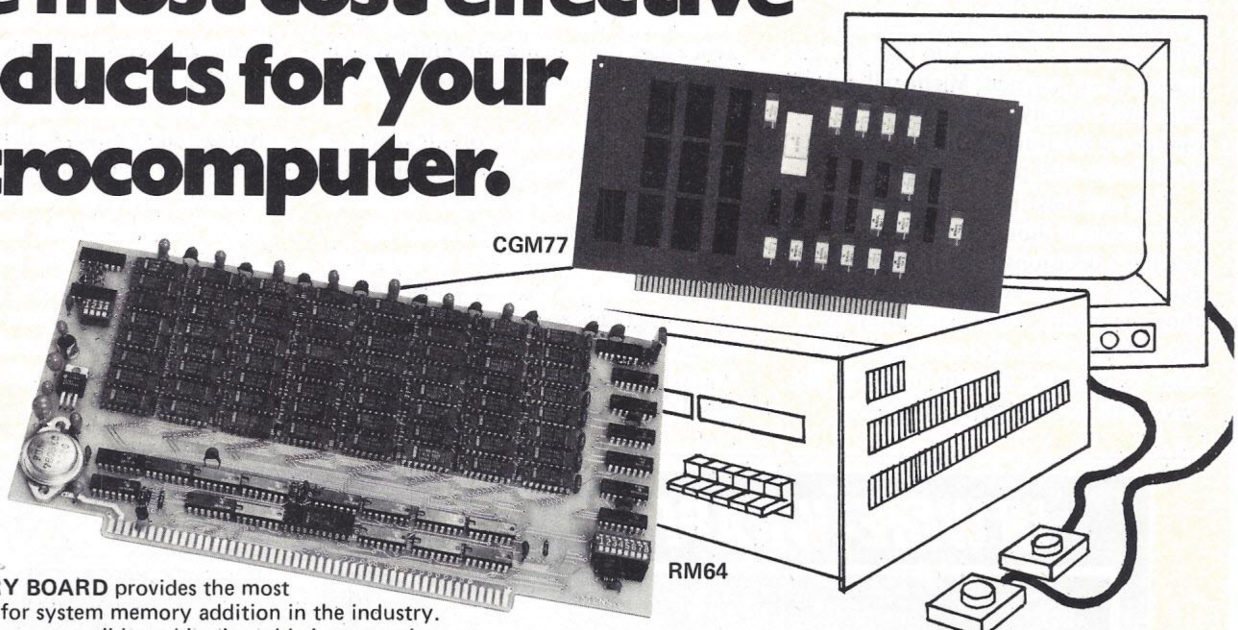
Schools now routinely keep official records of all major disciplinary actions – those on the level of, or more severe than, sending a kid to the office. This is of course done with an eye to the troublesome future; these things have a nasty way sometimes of winding up in court. It might be a good idea for a teacher to keep private records of more minor offenses, reprimands and punishments. Not necessarily out of reluctance that any sin should ever be washed away. No one will realize that Johnny raises hell only on days when that noisome orange glop is served at the snack bar, and his peculiar medical problem will not be brought to light, unless something more than ordinary human patience and thoroughness is employed in searching out unlikely-sounding correlations.

Or consider the case of a teacher thought by some to be biased against – or in favor of – some ethnic, religious, or other group among the students. The truth or falsity of the charges may be demonstrable in a statistical treatment of grades and disciplinary actions, by an administrator whose office or home is equipped for personal computing. It is not impossible that a parent or a bright high school student should attempt the same thing, given access to the records, and do it just as well or just as poorly. Will a school board – or a court, up in that continually troublesome future – accept this kind of home-grown evidence? Some member of the board may meanwhile crank up his own computer, and figure out a way to test the performance of the administrators, as well.

There is alas no law of nature that compels any of these adults to use the formidable powers of their machines for the benefit of people for whom the whole educational system is supposedly run. Time servers in teaching jobs can employ computers simply to make their own lives easier, and spend the saved time as entirely their own. Chances are they will not request a cut in pay because they no longer average eight hours' daily work. Computers will grade papers for them speedily, and help them do the easiest job of lesson planning they can manage – one size fits all. The parent who notices, one of these days, that multiple choice answer sheets are coming and going with the homework, might be well advised to do a little checking on their purpose. ■

Extensys Corporation.

The most cost effective products for your microcomputer.



THE RM64 MEMORY BOARD provides the most cost effective means for system memory addition in the industry. The board is S-100 bus compatible making it usable in over a dozen different microcomputer systems including ALTAIR and IMSAI.

It comes in three configurations: 32K, 48K or 64K bytes and is completely assembled, checked out and burned in for at least 50 hours prior to shipment. This complete testing procedure enables EXTENSYS to provide you a one year warranty on parts, labor and materials (assuming no misuse of the board occurs).

On board hardware is provided for:

- **Individual memory bank address selection in 8K increments;**
- **Complete dynamic refresh logic without loss of processing efficiency while programs are running;**
- **Board select logic which allows more than one 64K byte board per system;**
- **S-100 bus compatibility including on-board voltage regulator;**
- **Memory overlap which allows memory sharing the same address space to coexist in the same system; and**
- **Write protection in 16K blocks**

THE CGM77 COMPUTER GRAPHICS MACHINE is a fully tested system that provides the basis for developing the most sophisticated video graphics imaginable. The CGM77 board is S-100 bus compatible, making it usable on over a dozen different computer systems. The only additional components needed for a complete video graphics system are 8K bytes of RAM interfaced to the microcomputer system. The EXTENSYS RM64-32K RAM board is ideally suited for this.

Special control panels are available that provide universal control for interfacing with the graphics board. Output of the board is in the form of composite sync that can be used directly by a high resolution video monitor.

The heart of the system is the EXTENSYS GRAPHICS MACHINE. This is a sophisticated piece of software developed by EXTENSYS for the rapid implementation of video graphics. The GRAPHICS MACHINE permanently resides in PROM chips that are part of the board and therefore never need to be loaded. The GRAPHICS MACHINE software actually transforms the microprocessor into a much more powerful and sophisticated computer whose instruction set is ideally suited for the development of video graphics or video games.

Contact your local computer store or order directly from EXTENSYS

PC – 1

extensys corporation
592 Weddell Drive
Sunnyvale, California 94086

Please place my order for the following:

QTY	DESCRIPTION	AMOUNT
___	RM64-32K byte board @ $ 895 ea.	___
___	RM64-48K byte board @ $1195 ea.	___
___	RM64-64K byte board @ $1495 ea.	___
___	CGM77 computer graphics board @ $ 375 ea.	___
	Subtotal	___
	California residents add 6½% tax	___
	TOTAL	___

Please check method of payment:
- _ Check Enclosed
- _ BankAmericard No. ________ expiration date ________
- _ Master Charge No. ________ expiration date ________

Shipping and handling prepaid in continental United States

NAME ____________________

ADDRESS ____________________

CITY ________ STATE ______ ZIP ______ SIGNATURE: ____________

PHONE(INCLUDE AREA CODE) ____________

CIRCLE 15

HATE TO TYPE

word processing is for you

Writing may be easy but typing is hard. Most of us are not great typists, and would leave an ugly correction on a final typewritten letter rather than type the whole thing over again. Personal computer systems may be the salvation for the error-prone, two-fingered, "hunt-and-peck" typists.

If you are a frustrated typist, a computerized word processing system is for you. Good word processing systems take much pain out of composing letters and typing them. These systems let you throw your erasers away; you do not see a pile of wadded-up paper next to a person using a word processing system.

An Example

Basically, word processing involves an arrangement of machines and programs to make revision and typing of text easier. An example is the best way to explain this. Suppose you supervised a PTA drive to get people to donate used books to the school library. The book drive is over but you are faced with the huge task of writing to the 100 people who donated books. You have to do four different things with each letter:

1. Address each letter individually,
2. Thank the person by name,
3. List the books the person donated, and
4. List the value of the donated books, for tax purposes.

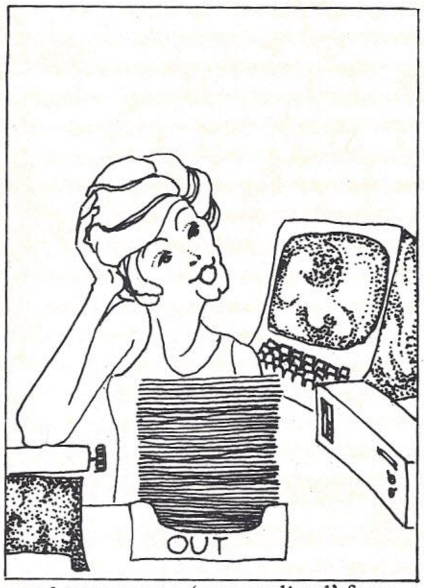
The book drive was not particularly well organized. You have the names and addresses of the people who donated books on one list; the names and the book titles are on another list. A third list contains the names and the donated value of the books, figured out by some accountant. None of these lists are in any particular order. If you had to sort out all that information and type 100 distinct and attractive thank-you letters, it might take you years. And you would still make mistakes on them.

This job requires a word processing system. So you talk a local home computer owner into setting up a Lemonade Computer Service Company to handle the task. The guy is not a typist but his simple personal computing system at home is hooked up to a good electric typewriter. He takes the lists and a pencil scratched copy of a short thank-you letter. You don't expect to see him again for a long time, so you start making up excuses to tell the PTA when they ask if the job is done.

Three days later, the Lemonade businessman gives you a stack of 100 beautiful, typewritten letters to sign and stuff into already addresses envelopes. That is the power of a word processing system.

An Inside Look

How does he do that job so fast? The trick is to let a computer do most of the work. The pencil scratched letter and the lists are entered into computer storage. The text is revised, the lists are double checked, and any corrections are made, always using the computer. Then the system puts the proper information with the proper name



and types out a 'personalized' form letter. The person using the system puts a sheet of paper in the typewriter and sits back to watch the typing of a perfectly accurate thank-you note.

A separate, equally simple operation is performed to type the envelopes. Since the names and addresses are already stored in the computer, no extra hand typing is necessary for the envelopes. The envelopes are even typed in the same order as the letters. Putting the letter and its envelope together is easy. The time saving for this whole process is enormous.

A word processing system requires four things:

First, you need a good electric typewriter. If you do a lot of work and speed is important, a printer is used instead of a typewriter.

Second, you need a computer and programs to handle the letter text, names, addresses and other data. Such computer programs let you make corrections, find specific phrases in the stored text and order the data by some method. These programs are called text editors.

Third, you need some computer storage. The text, names and addresses must be put where the computer can get at them easily and repeatedly. The best method of storage for a word processing system writes on and reads

from small, flexible disks coated with a magnetic material. Disk systems let the computer reach directly to specific information without fumbling through a long list of other information. Other methods of computer storage, such as magnetic tape, can also be used.

Fourth, you need a television screen connected to the computer. This can be either an ordinary TV or a special video monitor. This TV display saves paper and effort, because you make corrections only on the screen.

You must properly connect these four items, and it's these connections that make ordinary word processing systems expensive. The packaged systems vary widely in features and prices, running from about $5,000 to about $30,000. Businesses buy these expensive systems to save money.

Using A Personal Computer

A word processing system built around a personal computer can be cheaper than the commercial system with a plus: the computer can still be used for many other things. You may find the proper components and hook them up properly, or wait until somebody does this for you and offers a package for sale. You should not have to wait too long for this.

Some manufacturers are already selling good electric typewriters and printers that can be connected to personal computers. Several text editor programs are available. A computer, small disk storage system, suitable typewriter and text editor would cost less than $3000 today. This is still a bit steep for most homes but prices should fall. Naturally, if you already have a computer, TV and suitable computer storage, expansion to word processing is much cheaper.

Putting Everything In The Right Place

How does the word processor take messy lists and make sure the proper information gets in the proper thank-you letter? A special computer program is written just for this particular task. The program knows that each list has a name to which certain information is related. Each time the system starts a new letter, this special program selects the next name and searches all the lists for the additional information attached to that name. If the program can't find all the necessary information, it prints out an error message so the user knows what is missing. Then the computer goes on to the next name.

The text of the thank-you letter is stored in the computer, which has instructions to insert special, individualized information in certain places. For example, one of these places is at the beginning of the letter. Here the word processing system would insert the name and address. (This is the same information later used on the envelope.) An entire typical letter looks like Figure A, which has the specialized information highlighted and numbered.

Figure A) are from another list, under the same name. The $25.00 donation value (6 in Figure A) is off the third list. As a special touch, the Lemonade businessman added a fancy grammar feature. For more than one book title, the program automatically inserts an "and" before the last title and a comma between the other titles, if necessary. (See 4 in Figure A.)

Clever grammar in the letter text avoids inconsistencies in the tense of the verbs and the use of singular or plural words. Notice that the length of the sentence with the book titles is automatically adjusted by the word processor. No unsightly gaps appear if titles are short. Nothing is crowded together if many titles are listed. The system takes the space it needs and continues on. The date, address, sign-off and letter text are automatically and properly positioned on the paper. The program even centers the letter on the sheet of paper.

```
                    August, 1977

Mr. Alex Johnson
6276 Pine Street
Rightown,  USA  48018

Dear Mr. Johnson:

Thanks for your recent donation to the PTA
book drive.  The drive netted over 400 books
from 100 people like you, Mr. Johnson.

Your donation of Moby Dick, Star Trek (Volumes
I to XI) and The Illustrated Encyclopedia
(rare 1948 complete set) was especially welcome.

We have estimated that the value of your
donation is $25.00.  Please keep this letter
if you plan to take a tax deduction on this
donation.

                    Sincerely,

                    Book Drive Chairman
```

Here is how a word processing system would handle a typical form letter. If you think this looks just like any other personalized letter, then the system has done a good job.

On the "personalized" letter, the person's name has two forms (1 and 2 in Figure A). The shorter form, with the first name dropped, is used in the greeting and later in the body of the letter. The address (3 in Figure A) is from one list given to the Lemonade businessman. The book titles (5 in

Text Editing By Computer

The original lists are entered into the computer system initially using a text editor program and the computer keyboard. This is the first and only time the majority of the information is hand typed. If you are a good programmer you may not need to buy the text editing programs. You can write them yourself. Text editors can be simple or complex. They can keep track of the words, sentences, paragraphs and pages of the text. The amount of text they can store depends on the size of your computer system, especially the computer storage part.

Once you have typed the original text and lists, mistakes and all, into the word processing system, you never have to retype them again. The text editor will display the words on the computer system's TV screen. If you see a mistake, or want to make a change, type the appropriate command, and the text editor makes the correction. The mistake disappears and the correction appears. No paper is wasted and no erasers are worn. When the text is fully corrected, it is typed by the computer-run typewriter quickly and without a mistake, on paper.

The text editor keeps track of how the final version looks. If you insert a new sentence, all the other text is shifted to accomodate the new text. Paragraphs and pages are adjusted and the new version is shown on the screen. For long text, you always know how much space you will need and where every page starts and stops. All this is done automatically, regardless of the number of corrections you make.

The ability to find a particular word or phrase is a useful feature of a text editor. Suppose the household cook has recipes stored in a word processing system and wants to find recipes that use leftover mashed potatoes. The cook asks the text editor to find the words "leftover" or "mashed potatoes" or simply "potato" among the information stored in the word processing system. Every time those words appear in the stored recipes, the system prints the surrounding text and allows the cook to examine the recipe.

Of course, if you have misspelled "potato" as "patoto" when you stored a certain recipe, the text editor will not find that particular recipe. Similarly, if you want to find the word "for" in the text of a letter you have stored, the system would find things such as "information", "format" and "therefore" as well as "for". The text editor is obediently doing exactly what you asked in both cases. It has no way of recognizing your mistakes. (Some systems allow you to get around this "for" problem with additional instructions.)

Typical Cost for Setting up a Minicomputer Word-Processing System

Although systems are offered for appreciably more and somewhat less, the setup described in a news release from The Software Store is representative.

"An inexpensive word/text processing system, currently operational on MITS Altair equipment, is now available. This Mini Word Processing (MWP) System does the following:

1. processes letters, documents mailing lists;
2. provides operator prompts to minimize training time;
3. supports maintenance of name and address files;
4. allows use of all the EDIT functions of BASIC;
5. arrives with complete documentation and is available now!

The price for the system, delivered on a diskette and with a complete user's manual, is $150.00."

Hardware required includes an Altair 8800a or 8800b, 28K of memory, 1 floppy disk drive, 1 CRT and 1 printer. "Upper and lower case letters are hardware dependent." (That is, if your printer offers both, the system will offer both.)

Now, roughly, the computer will cost about $1K, the 32K memory board about $1.8K, the floppy about $2K, the CRT (better be 80 characters) about $.3K, and the printer about $1.2K (assuming that it's a modified Selectric with an interface to this system, not fast, but neat. Including the software package, this comes to about $6500. As the author of *Hate to Type?* remarks, you can do a lot of other things with the computer as well.

Further, you can select other systems, wheel and deal, buy more, buy less, do your own, lie, cheat and swindle to acquire a system for less. However, a working, debugged, ready-to-go system must cost something. This is an example, and a rather attractive one. The Software Store is at 706 Chippewa Square, Marquette, Michigan 49855.

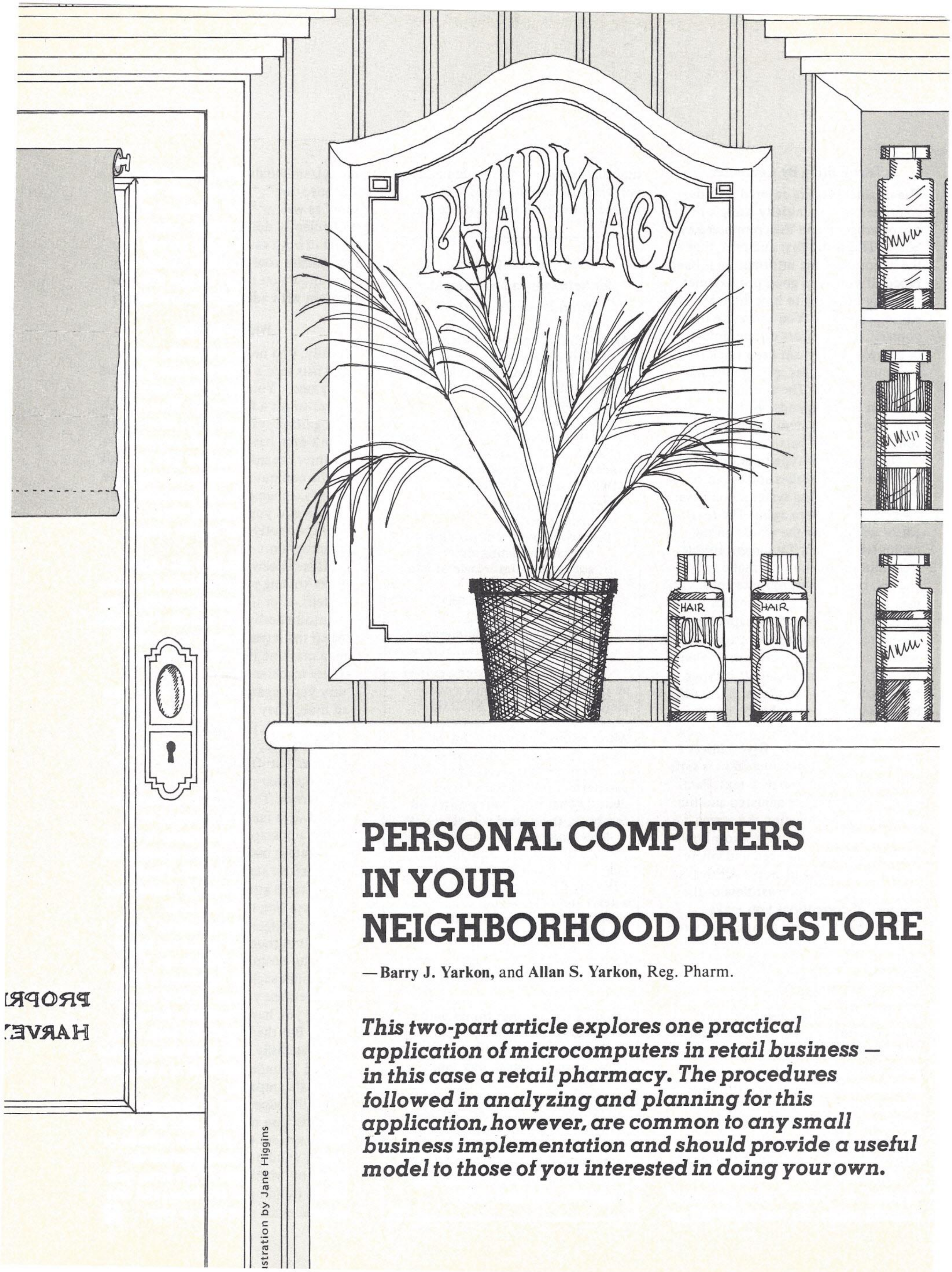
What's Coming

Ideally, you need to type the words and lists into a word processing system only once. You also never waste any paper to get a final copy. When speech recognition systems are perfected, you won't even have to touch a typewriter to enter the information. You just talk to the computer. With good computer speech recognition, the computer will understand your spoken words, spell them correctly and use the proper punctuation.

It is conceivable that the whole letter writing procedure could be automated. After the text is entered, paper is automatically fed into the typewriter. When the letter is done, it is grabbed by a machine that signs your name. These machines can be adjusted to vary your signature slightly from time to time. They can even be equipped with colored, felt-tipped pens. After the signature, the machine folds the letter and stuffs it into the addressed envelope that just came out of the typewriter. The envelope is machine-sealed. It is machine-stamped with either a postage meter stamp or a regular stamp; maybe even a special commemorative stamp is used. Finally, the letter is automatically routed to a bin according to its zip code.

This perfect "personalized" form letter is untouched by human hands until the postman picks it up. You may think such an operation is horrible because you are tricked into believing you have received a personal letter. But the lowly PTA book drive could as easily have been the city-wide United Foundation drive or a national political campaign.

As the supervisor of these giant operations, you might be glad to have such a system to use. Your opinion of a good word processing system depends on which side of the letter you are on (sending or receiving) and also on how much you hate to type. ■

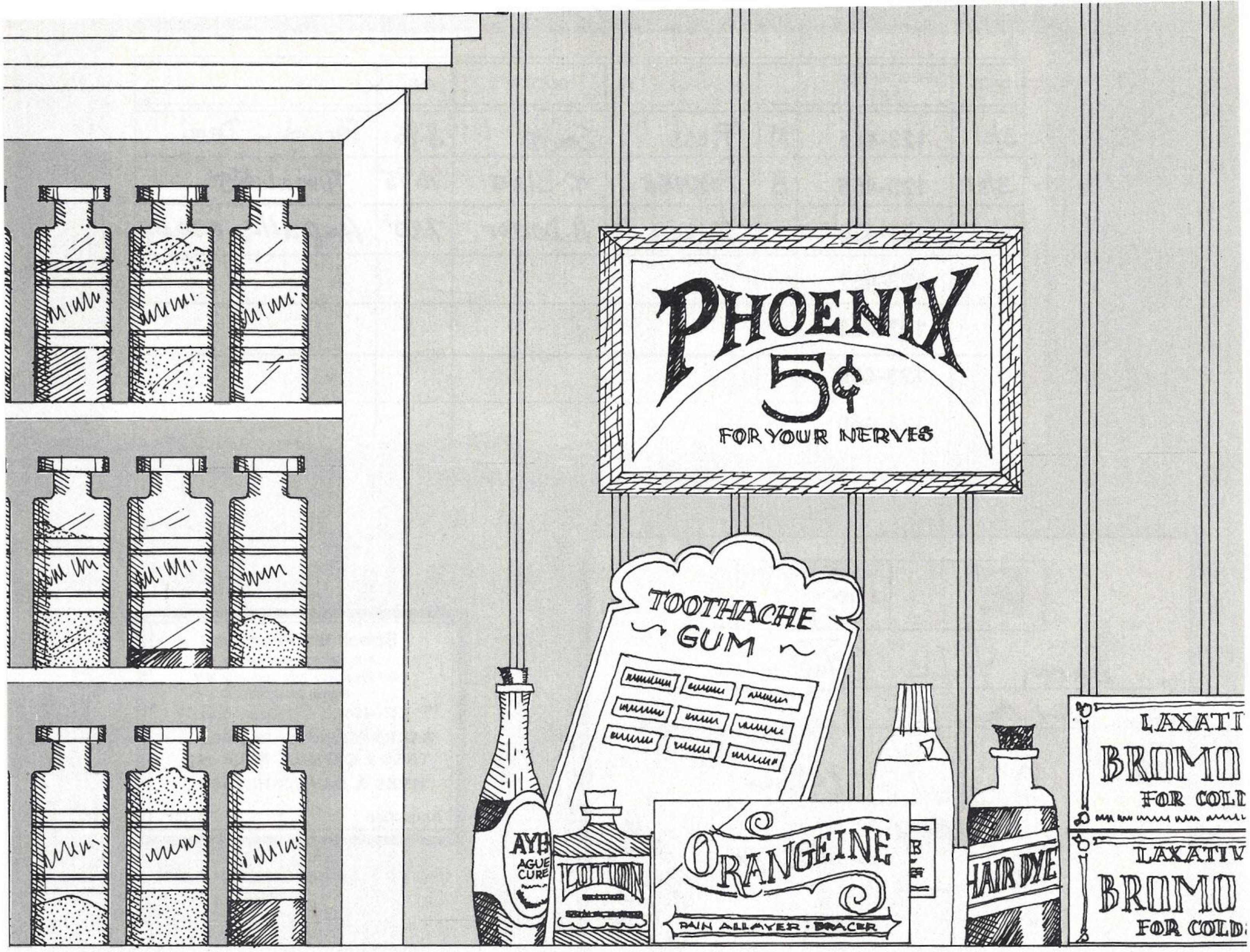


stration by Jane Higgins

PERSONAL COMPUTERS IN YOUR NEIGHBORHOOD DRUGSTORE

—**Barry J. Yarkon,** and **Allan S. Yarkon,** Reg. Pharm.

This two-part article explores one practical application of microcomputers in retail business — in this case a retail pharmacy. The procedures followed in analyzing and planning for this application, however, are common to any small business implementation and should provide a useful model to those of you interested in doing your own.



To install your own microcomputer system you must tackle three basic areas:

1. Define your Application

The successful application of any computer (in any business) depends on detailed knowledge of the job to be accomplished. You have an edge over an outsider when the system to be designed is a familiar application—but you will be forced to view the task in a more critical and objective way. Study the system as it presently exists, observing what each person does and how. Collect samples of all existing forms and documents. Then work up a plan which takes these into account. Your plan must be detailed in proportion to the complexity of the proposed implementation. It should cover necessary business forms and training information as well as equipment required; and, account for the expected return on your investment. Often this analysis will bear a dividend even should you not carry through due to the time or money required! You'll be glad you thought things through.

2. Set Up a Physical System

Based upon your analysis, you must now select, acquire and then assemble the components which form a minimum system. This may involve only sufficient hardware to begin programming, or, if you are more sure of success, a basic configuration of the system. In selecting hardware, be sure that you will receive sufficient technical support in areas which you are not completely qualified (i.e., in troubleshooting, interfacing, etc.) Also, make sure that any special-purpose components are in fact available—this definitely includes the software operating system you will be using on the hardware. Don't purchase anything until you are certain that you understand the function of all components as well as the effort necessary to learn and apply a particular operating software package. Comparison shop.

3. Program your Application

At this point, with your equipment assembled (and tested) and an understanding of how to use the system software, work can begin on your application software. While it is beyond the scope of this article to cover general programming techniques, experience has shown that the best tactic is to break the job down into various small, managable tasks to be taken one at a time. Be sure to document each portion sufficiently to recall how it functions at a later date.

Where to Start

Now let's follow through with an actual implementation. Take a look at a typical retail pharmacy

DATE	PRESCRIPTION NUMBER		PATIENT	DOCTOR	AMT.	ITEM:
3/31	123-454	N	Ross	Smith	3.95	Percodan Demi
3/31	123-455	B	Frange	T. Blue	10.95	Tuinal 5gr.
3/31	123-456		Yarkon	A. Doctor	7.20	Ampicillin 250mg
	123-457					
	123-458					
	123-459					
	123-460					

Figure 1. *Rx Transaction Ledger.*

PHONED 10 A.M. P.M.		DELIVER 2 PM	WILL CALL

FOR Barry Yarkon YARKON
ADDRESS 69-1 Main St.

Rx
Ampicillin 250mg
#24
Sig: i QID prn

123-456
3.31.77
7.20
ASY

REFILL 1 TIMES
BNDD NO. DR. A. Doctor, MD.
DATE 3/31/77 231 Rnd. Swmp ADDRESS

Figure 2. *Typical Prescription Order (Rx).*

Specimen Pharmacy
A.S. Yarkon—B.J. Yarkon
1457 Broadway, New York City, N.Y.
Phone: 221-6093
No. 123-456 Date 3.31.77
BARRY YARKON: 79-5 Main
TAKE 1 CAPSULE FOUR (4)
TIMES A DAY AS NEEDED.
Ampicillin Dr. A. Doctor

Figure 3. *Sample Container Label.*

from the pharmacist's side of the counter. While a retail pharmacy usually provides sundries and related goods and services, we shall confine our examination to prescription dispensing.

The prescription area is the direct responsibility of a Supervising Pharmacist and may include other dispensing pharmacists and/or interns (trainees). In each state, a Board of Pharmacy establishes the rules by which the pharmacies it licenses must operate—particularly their record-keeping. Certain records are also required by Federal agencies. [Interestingly, it was only as recently as 1976 that a Task Force on Innovative Pharmacy Care first established standards for electronic prescription systems. Prior to this, it had been questionable whether a prescription in electronic transmission or in computer storage was even valid in the legal sense!]

When you hand in a prescription order the pharmacist usually enters it into a ledger (see Figure 1) with at least:

Rx number	Prescriber's name
Date	Price.
Patient's name	

On the prescription itself (Figure 2), at least:

Rx number	Patient's address
Date	Dispenser's initials
Patient's name	Price.

He (or she) then types a container label (Figure 3) with at least:

Rx number	Directions for usage
Date	Prescriber's name;
Patient's name	

and, sometimes such other information as drug name, refill(s) allowed and cautions for usage or storage.

In many pharmacies the dispensing pharmacist will retrieve, update and refile the patient's Family Drug Usage record (see Figure 4). He usually scans this history for likely complications with drugs already being taken (called drug-drug interaction). The well done family record will also contain known drug allergies which the pharmacist checks as well. Lastly, the proper drug in a proper dosage strength is counted/measured/or manufactured, labeled and given to the patient.

YARKON	BARRY	J
LAST NAME	FIRST NAME	M
69-1 Main Street, Upstairs		
ADDRESS		
Flushing,	NY	10303
CITY	STATE	ZIP
212/ 456--7890		
PHONE	UNION PLAN	

AGE	PATIENT	ALLERGIES
32	Barry	
30	Beverly	
6	Rachel	Steroids
4	Deborah	

DATE	DOCTOR		PRESCRIPTION NUMBER	PATIENT	AMT	
6/3/76	H. Skelton		099--509	Deborah	4.25	Benylin Expect.
8/1/76	H. Skelton		111--798	Rachel	8.95	Illosone 125 mgm
3/31/77	A. Doctor		123--456	Barry	7.20	Ampicillin 250 mg

Figure 4. *Typical Family Drug History Record.*

DATE	DOCTOR	R. PH. REFILL TIMES	PRESCRIPTION NUMBER	PATIENT	AMT
3.31.77	A. DOCTOR	ASY 1	123-456	B. YARKON	7.20

FOR INCOME TAX AND/OR INSURANCE REPORTS ETC.
THIS IS YOUR PROOF OF MEDICAL EXPENSE

SPECIMEN PHARMACY

1457 BROADWAY NEW YORK CITY, N.Y. 10036

Phone For Service

221-6093

Prescriptions Called For And Delivered
OPEN 365 DAYS A YEAR

WE MAINTAIN A COMPLETE RECORD OF ALL YOUR FAMILY'S PRESCRIPTION PURCHASES

Figure 5. *Specimen Purchase Receipt.*

AMPICILLIN

Unless otherwise directed, this medication should be taken on an empty stomach, *or* one hour before meals, *or* 3 hours after meals. It should *not* be taken with fruit juices.

Reminder courtesy of:

SPECIMEN PHARMACY, N.Y.C.

Figure 6. *Drug Information Leaflet.*

The patient will often receive a detailed medical receipt for income tax purposes (Figure 5) which contains at least:

Rx number	Drug Item
Date	Prescriber's name
Patient's name	Price.

Prescriptions which qualify for private or governmental reimbursement plans require further paperwork for the pharmacist or his staff. Another professional service often rendered by retail pharmacies is a preprinted Drug Information leaflet which explains the nature of the item dispensed (see Figure 6).

It is clear that much of the data is entered repetitively and is therefore subject to multiple transcription and filing (not *filling*) error. So-called "one-write" carbonized forms can simplify this data entry procedure somewhat—a microcomputer system, though, has the inherent capability for automation and control unavailable to manual procedures. The pharmacy's record-keeping problems are commonly found in many other small businesses as well.

Establish a Plan

Figure 7 shows a graphic outline of the areas related to our overall plan for applying a microcomputer system to retail pharmacies. The key to this system is obviously to capture and make multiple use of data contained in the prescription order received from each patient/client.

By entering some information in addition to that usually typed onto the container label, alone, we can cause a microcomputer system to use this data to:

- Fill out an error-free container label
- Issue a proper receipt
- Search the Family Drug History—and update it
- Print an entry to the daily transaction ledger
- Decrement an inventory of ethical drugs by the amount of the current prescription.

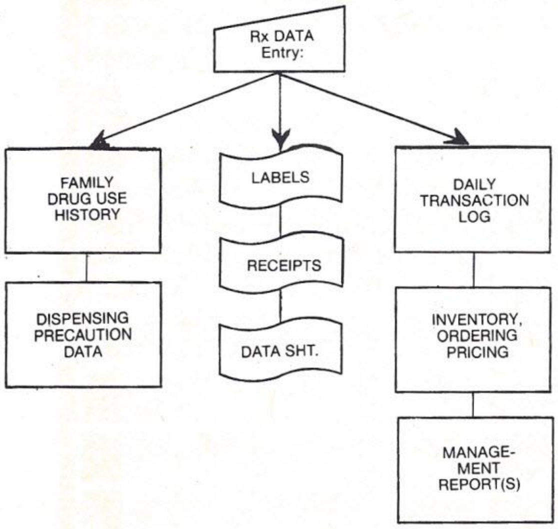


Figure 7. *Organization of pharmacy support operations.*

Additionally, potential problems could be flagged via the history portion of the family record; short supplies can be flagged in inventory with an order compiled for the drug wholesaler; and, periodic or on-demand managerial reports could be accomodated showing such things as number of new and refilled Rxs, dollar values of same, turnover ratios of drug products, sales volume analysis, profit and loss, etc.

A microcomputer could free the retail pharmacist of many purely clerical details allowing him to concentrate on professional and administrative functions while increasing the quality of service to his clients.

Beyond the Plan

You will find among most business people a profound lack of knowledge of how computers work, or even what they can do. The pharmacist is no exception, so, this pharmacy system must be carefully designed around his skill level. It must be as straightfoward as possible to understand and use. Since the "psychology of acceptance" revolves around removal of roadblocks, the system must:

- require a minimum of new terms;
- have quickly comprehended procedures;
- clearly step the user through required dialogs and interractions; and,
- *not* require special skills or knowledge beyond the application.

A conversational (interactive) program running on the system hardware would be an ideal training vehicle. This would have the double benefit of enabling the system's user to train new users (employees) himself. User inputs can be monitored for improper responses.

The cornerstone of this system's plan is the man/machine dialog. We have found it useful to set up a "storyboard" much as artists or filmmakers do to plan their scenarios graphically. In this manner one can quickly develop and polish the user/system dialog *before* formal coding begins. Don't fall into the trap of allowing coding or language peculiarities to obscure this dialog—it is the only part of the system a naive user can recognize.

In these illustrations, the bolder type on the righthand side represent the user's typed responses. Upon powering up the system, only authorized users aware of the password can "unlock" the program:

```
GO
PLEASE IDENTIFY?                 XYZ
IDENTIFY!!                       BJY

*UNLOCK & INITIALIZE:

ENTER DATE?                      3.31.77
STARTING RX NO.?                 123456

*NEED A MENU?                    Y
```

"Yes" to the last clears the terminal's screen and displays this function directory:

```
*OPERATIONS AVAILABLE:

NEW    - ENTER A NEW RX
REF    - REFILL AN EXISTING RX
LA     - PRINT A CONTAINER LABEL
REC    - PRINT A RECEIPT
REV    - REVIEW CURRENT RX
S      - SAVE DATA FOR NEXT RX
K      - KILL THIS ENTRY
$      - PRINT A MANAGEMENT REPORT

? NEW
```

The "?" requests an operation to be done. Let's look at the dialog around new prescriptions (NEW):

```
*ENTER RX 123-456:

DISP. PHARMACIST?                ASY
PATIENT LAST NAME, FIRST?        Yarkon, Barry
STREET, CITY?                    1457 Broadway, NYC
PATIENT'S AGE, SEX?              32, M
PHYSICIAN'S NAME?                A. Doctor
CONTROL?(N,B,O)                  O
RX ITEM, STRENGTH?               Ampicillin, 250mg
QUANTITY, DOSAGE FORM?
  (T,C,S,P,CRM,UNG)?             24, C
SIGNA?                           1,QID,prn
CAUTIONS (N=None)?               N
REFILLABILITY?                   1
PRICE (IF KNOWN)?                7.20
*END OF RX ENTRY
```

CONTROL? denotes special requirements for narcotic (N), barbiturate (B), or general (O) drug items. Controlled drugs require a prefix or suffix for the ledger number—so, an 'N' or 'B' will be printed for the current Rx record.

DOSAGE FORM? is an abbreviation for: tablets(T), capsules(C), suppositories(S), powders(P), creams(CRM), ointment(UNG, *unguent*).

SIGNA? is a Latin term for directions which are often written in a traditional latin shorthand on each prescription by the prescriber. For example, this one is:

one/four times a day/as needed

There is a large list of such terms and it is most important

to allow the pharmacist to transcribe the signa from the prescription into the system—*he* understands them as a matter of course. This translational ability might be carried further by designing your applications software to translate the signa into label directions in some other language altogether (i.e., Spanish, French or German).

Refer to screen illustrations 4-6 for examples of a container label, a receipt and a simple management

```
*SPECIMEN LABEL

NO. 123-456                                DATE: 3.31.77

Barry Yarkon
1457 Broadway                                    $ 7.20

TAKE ONE (1) CAPSULE FOUR TIMES A DAY AS NEEDED.

Ampicillin              REFILL: 1          DR. A. Doctor

*END OF LABEL
```

```
*SPECIMEN RECEIPT

FOR INCOME TAX AND/OR INSURANCE REPORTS, ETC.
THIS IS YOUR PROOF OF MEDICAL EXPENSE:

DATE: 3.31.77                    AMT: $7.20

PATIENT: Barry Yarkon
RX NO.:    123-456               DR. A. Doctor
R. PH.:    ASY                   REFILL: 1

              SPECIMEN PHARMACY
         1457 Broadway, Suite 305, NYC

*END OF RECEIPT
```

```
*MANAGEMENT REPORT

DATE: 3.31.77                          PER: ASY

LEDGER NOS.: 123-455   TO 123-456

                       QUANTITY:   DOLLARS:
NEX RX                     1          7.20
REFILL RX                  1          4.95
                       --------    --------
   TOTALS—                 2        $12.15

*END OF REPORT
```

information report. Notice how the label instructions for use are exploded. Label and receipt require no further typing by the pharmacist.

Hardware and Software

In the second installment of this article we will detail the system operating software [a disc-based extended BASIC] and hardware components required to effect this retail pharmacy system. We will look more closely at criteria for designing and maintaining diskette-based files, and discuss a unique multi-station configuration allowing *two or more pharmacists to dispense at the same time.*

Meanwhile, why not begin your own pet implementation of a retail microcomputer system. Begin by defining and analyzing. . . the fun is in the doing.

Personal Computing Comes to

A major exhibition on computers for home, personal, business and school use will be held in Chicago on October 27, 28 and 29, 1977.

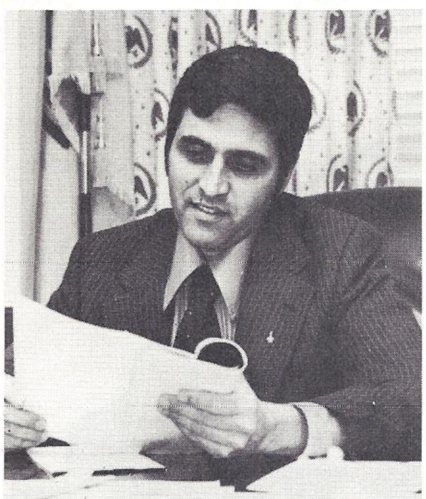
Sponsored by Personal Computing magazine, this show will bring 100's of exhibitors of consumer computer products. It will be held at the world's largest Holiday Inn on the Kennedy Expressway near O'Hare International Airport.

Dozens of seminars are planned to teach computing concepts to the general public. There will be banquets, workshops, a computer school--100's of doorprizes including a personal computer as the grandprize.

This show is the first major personal computing event to be held in the Midwest. Plan on attending now. For free \$1 discount tickets and more information write: **Personal Computing Show/401 Louisiana SE/Suite G/ Albuquerque, NM 87108.**

A Personal Computing Interview

Senator Harrison Schmitt on COMPUTERS AND TELECOMMUNICATIONS



Harrison Schmitt was the first civilian scientist/astronaut to go to the moon, and has recently become the junior senator from New Mexico after a long tour of that large state in a red pickup truck, which now graces the Senate parking lot in Washington. This interview took place in April, 1977.

Personal Computing: *Have you been aware of the great activity in personal ownership and use of computers?*
Senator Schmitt: Yes, I've been reading various articles in newspapers and hearing about it, but it's also been mentioned in the hearings of the Communications Subcommittee in Congress. It's part of the total discussion of the telecommunications revolution. One of the areas where there is great future potential is the possibility of communication by computers, much the same as we now have voice communication. It is one of those things that any national policy we might devise should take into account. The system ought to be compatible with two-way computer communication. Suddenly the barriers are gone. The barriers between voice and radio and television and computing were there for a while and now they're disappearing. There are no clear barriers between various modes of communication and we've got to consider that.

Did you have to learn much directly about computers for the Apollo mission?
I had to learn computer operations, because we had small computers in the spacecraft. We needed to know basic computer theory, because we had to be sure we didn't louse up the computers. That was a confidence-building thing as well as learning how to make the system operate in an appropriate way.

Does it feel odd now to realize that you bet your life on computer systems that were designed in the early sixties and now seem quite old-fashioned?
Well, we did bet our lives on computer technology, but the Apollo systems were designed to do a particular job and they did it. Using a computer to fly things is fairly new and even I learned to fly aircraft and grew up in a time when computers were a sort of novelty. You have to learn to develop that kind of confidence.

Do you use, or would you like to use, computers in your daily personal life?
Very definitely. We're setting up a computerized system in my Senatorial office in Washington to cut down the amount of time required to handle correspondence, providing good answers and services to constituents. The volume of communication from the public has been increasing and I think that's a good thing, but the Senatorial mail system has yet to adapt fully to the fact that it's happening. We've set up a procedure that is compatible with a computer-based system as soon as they can supply that to us. Unfortunately, being ninety-ninth in seniority, we're a little bit down

on the list. We expect to have it by the end of the year.

A personal computing system might allow you to do other tasks like little statistical studies or organization of research material. Do you have access to such a system?

We don't at the present time. We don't have an allowance for that, so if we took it on, we'd have to be sure there was some way to finance it.

A magazine like ours might be able to persuade a manufacturer to provide a system for your use. Their pay would be the information on how the system can be useful to an elected legislator, with any programming that might be developed. The magazine would be delighted to have the interesting editorial material deriving from it.

Whatever manufacturer provided that, it would probably be considered a gift. I think it would be of great interest to have the capability, whether or not it would work out in our system. I'm on the Ethics Committee, and I think we'd have to pay for everything we did . . . which would be fine. We're going to consider that as we get more and more familiar with what we're going to do in our Senatorial offices.

Are you working on privacy legislation?

I think it's going to be a very tough issue and one of my standard questions in dealing with the national telecommunications system has been: are you confident that we can ensure individual privacy as we move into the age where there are going to be fund transfers, record transfers and this kind of thing via computer? You get a variety of answers. I'm not confident yet that we know how to do it. However, being a technologist of a sort, I feel that we can probably work this one out if we just don't try to rush into it.

Is there a likelihood of legislation that will require all computers to be licensed?

I have not detected it. I think the pressure for such licensing would start to increase if we did not find a way to ensure privacy in the distribution of information.

The rise of personal computing has brought on a lot of talk about ro-

bots and artificial intelligence. Do you think there's anything to that?
Well, I think so. I think we're going to see more and more use of so-called artificial intelligence for space and certain hazardous jobs here on Earth. I don't think it'll ever substitute for Man's "being there." Man's wanting to be somewhere to do certain things is more a psychological desire than it is what we would call a cause and benefit term, a practical desire. It has its own value, a psychological value. The capability of the robot to make a certain spectrum of judgements would be limited by Man's imagination. I can conceive of programming raw judgemental capability into a robot, where the robot might make judgements that you never anticipated.

As one of the few men who have been to the moon, have you also read much science fiction?
As a child . . . in high school, I read a lot of science fiction. But I do it very little, very selectively now, because of the circumstances and time.

We are biased, of course, in believing that the spread of personal computing will be of help to the private citizen in a big way, equalizing the difference in power between him and the institutions. Does that sound reasonable? The ordinary complaint is that computers depersonalize people, treating them as meaningless numbers.
I don't know how much equalizing it will do. I don't seriously think that computers will homogenize the population. The thing that keeps the population from being homogeneous is the difference in capabilities of the people. You give a computer to somebody and they'll use that computer based on their abilities, rather than on any set program. The personal computer will probably maintain, if not enhance, individual capabilities in our society. Whether or not it gives the individual a better chance to compete with institutions, I don't know. Unfortunately, governmental institutions are expanding at such a rate, with associated complexity, that I think it's going to be very difficult to keep up with it, even with a computer.

I'm really getting very concerned, if not pessimistic, about the ability of individuals in this country to continue to have the desire to compete with institutions. We are making ordinary activities in this world of ours so difficult as a consequence of regulations by faceless bureaucrats who cannot be thrown out of office by the electorate.

"I don't seriously think that computers will homogenize the population . . . the personal computer will probably maintain, if not enhance, individual capabilities in our society."

What do you think is important that we've overlooked in this interview?
I wish that you had asked more about the national telecommunications system that we are now discussing in the Congress. That's national with a small "n." How do you define the system that we are going to need in the future? Is there an area within this system that you could define as a natural monopoly? How do you find the boundary between such a monopoly and a purely competetive industry?

I've been struggling with my own definition and I've got some ideas. I'm attracted by the analogy that a national telecommunications system is much like a highway system. Computers, like people, need a highway over which they can communicate. That probably needs to be regulated in some way so that the user is confident that he's getting a low-noise, high quality highway.

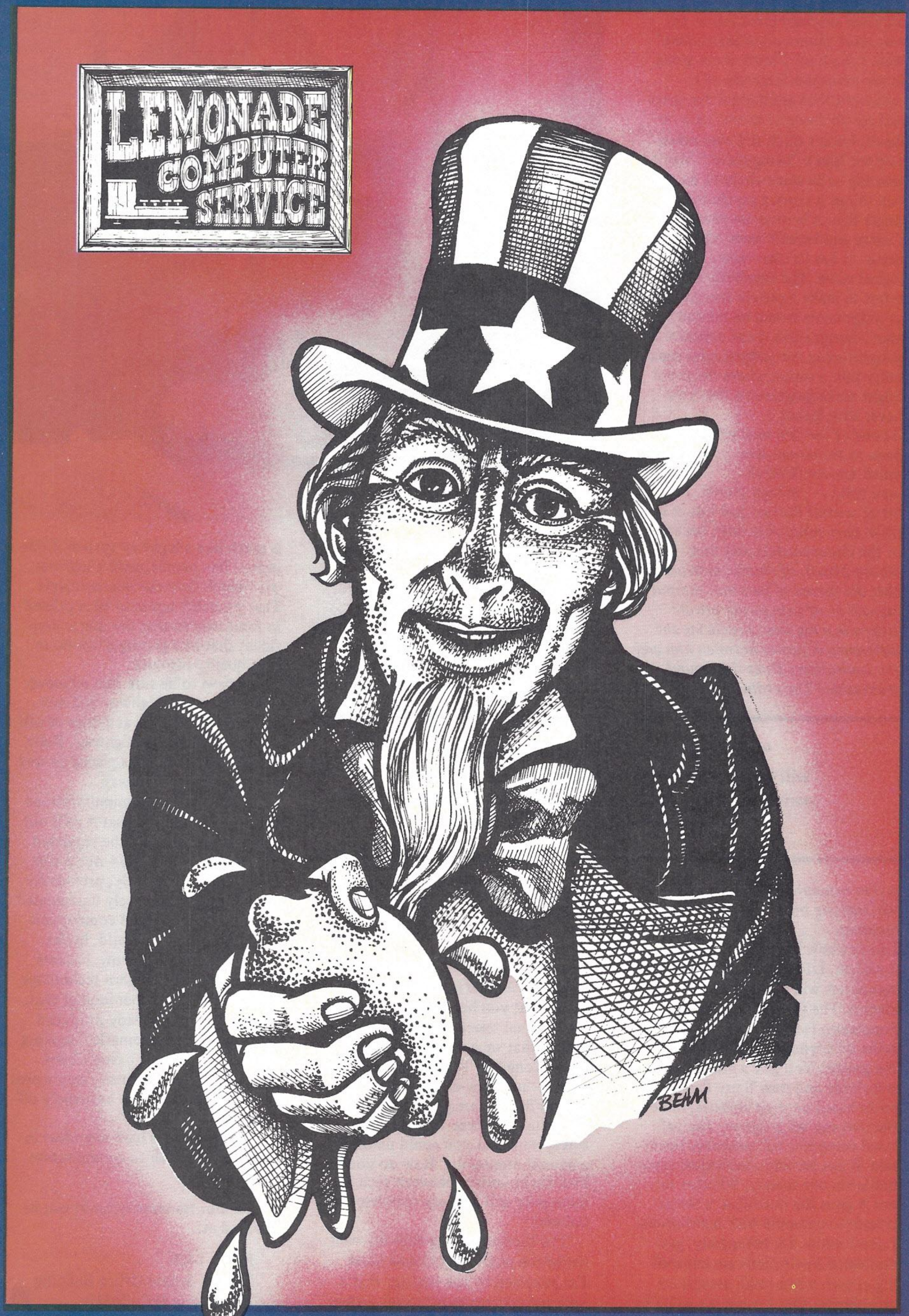
We have to be sure that whatever we stick on that highway will not seriously disrupt the activities of other users of the same highway. There are certain restrictions placed on any car you put on a freeway.

I hope the next communications act is not so specific that it calls out everything that has to be regulated. The beauty of the 1934 Act is that it did, up until recently, provide for a fairly strong, relatively unregulated system that is the best communications system in the world, by far. We're starting to see now, because of the advent of modern technology, a need to take another look at that. In fact, we have already produced what we call a "schematic," looking at the system, but I haven't been able to put that schematic into words yet; it's going to be an extremely difficult task.

If your readers have any ideas in this area and wish to communicate with me . . .

Senator Harrison Schmitt can be reached at:

Room 1409
Dirksen Senate Office Building
Washington D.C. 20510


LEMONADE
COMPUTER
SERVICE
BEHM

TAX ASPECTS OF PERSONAL COMPUTING or

WHEN IS A HOBBY NOT A HOBBY?

by Kenneth S. Widelitz

This rather cynical and extremely useful view of Lemonade Computer Service activity is offered by an attorney with a shiny new office on Wilshire Boulevard in Los Angeles. Ken Widelitz first delivered this information in a talk at the West Coast Computer Fair and his text appears in the proceedings of that gathering. This treatment, revised in response to interested reaction, should be of special interest.

The title of this article could be "How to Get Uncle Sam to Subsidize Your Microcomputer." As the present title suggests, you must use your microcomputer in entrepreneurial fashion to obtain any of the benefits discussed. However, you needn't be an overly succesful entrepreneur to obtain a subsidiary.

The Internal Revenue Code allows benefits to taxpayers who are "engaged in an activity for profit," as opposed to those who are not engaged in an activity for a profit (a hobby). For example, by qualifying an activity as one engaged in for profit, the microcomputer entrepreneur will be allowed an investment tax credit equal to 10% of his investment in his microcomputer and other assets used in the activity. Further, such assets can be depreciated. It is also possible to deduct a portion of residence expenses if you operate a Lemonade microcomputer business out of your home or apartment.

If your Lemonade application is determined to be one engaged in for profit, then you are entitled to substantial tax benefits. Of foremost interest to the microcomputer user is the investment tax credit; an investment tax credit equal to 10% of the purchase price of tangible personal property which is used in a trade or business is allowed. The theory behind the investment tax credit is to encourage investment. For example, if you purchase a microcomputer and peripherals for $8,000, you are entitled to an investment tax credit of $800 in the year of purchase (see Chart 1). The $800 is a credit and not a deduction. It is subtracted from the tax liability after the tax liability is determined.

the cost of hardware

In order to qualify for a full investment tax credit, the asset on which the investment tax credit is taken must be an asset which has a useful life of seven or more years. A seven year useful life for a microcomputer is reasonable and falls within the asset depreciation range guidelines for useful lives established by the IRS. If the asset has a useful life of more than three years but less than five years, only one-third of the full credit is allowed. If the life is more than five years but less than seven years, two-thirds of the full credit is allowed.

Another tax benefit is that the microcomputer may be depreciated. Depreciation is a theoretical slush fund which is set aside to replace an asset when it is worn out and useless. Even if the funds are not actually set aside in a bank account or elsewhere, the IRC (Internal Revenue Code) allows the slush fund to be deducted as if it were being stashed away. The amount of such depreciation is deducted from income to determine the tax liability (as opposed to the investment tax credit which is subtracted once the tax liability is determined.) There are a number of methods to determine the amount of depreciation in a given year. The simplest is straight line depreciation and for purposes of example, the straight line method will be used. The asset must be depreciated over the same life assigned for puposes of the investment tax credit. In the example the microcomputer and peripherals are purchased for $8,000 and have a life of seven years. Assume the system will have a salvage value of $1,000 at the end of seven years. Depreciation in the amount of $1,000 per year may be deducted (see Chart 1).

software

The cost of software may be deducted currently as an expense. However, if software is purchased as a package with hardware, it must be depreciated over the life of the hardware.

other expenses

Other Lemonade expenses which may be deducted include telephone, trade

journals (Personal Computing, etc.), postage, stationery, mileage at 15¢ per mile, wages of employees (if any), and perhaps an amount for a portion of your residence.

To deduct as an expense a portion of your residence costs, the portion of the residence to be deducted must be used exclusively and on a regular basis as the principal place of business. This requirement was added by the tax reform act of 1976 and is strictly construed. Theoretically, if you play Star Trek with your friends on the computer in your den and also use the den for your Lemonade business, you lose the deduction. If your residence is an apartment, a pro rata percentage of rent, electricity, gas, etc., may be deducted (see Chart 1). If your residence is your house, the formula is a little bit more complicated because some expenses such as interest on your mortgage and real property taxes are already deducted.

☆CHART 1

CALCULATION OF BENEFITS

INVESTMENT TAX CREDIT

Computer and Peripherals; Assume Seven Year Life

Cost	$8000	
Investment Tax Credit (10%)	800	

DEPRECIATION

Computer and Peripherals; Assume Seven Year Life Straight Line Depreciation

Cost	$8000	
Salvage Value (arbitrary)	1000	
Total Depreciation Allowed		$7000
Depreciation Per Year ($7000/7 Years)		1000

PRO RATA RESIDENCE

Assume One Of Four Rooms Used Exclusively and Regularly For Business, Pro Rata Portion 25%

Yearly Apartment Rental	$3500	
Yearly Gas	180	
Yearly Electricity	320	
Total Residence Expenses		$4000
Pro Rata Portion (25%)		1000

☆CHART 2

ADULT ENTERTAINMENT COMPUTER COMPANY
Five Year Forecast
IRC Section 183 Presumption, Election To Postpone Determination Filed

	1977	1978	1979	1980	1981
Gross Income	$ 0	$1000	$2000	$4000	$6000
Expenses					
Interest	800	800	800	800	800
Trade Journals	100	100	100	100	100
Office Expenses	200	200	200	200	200
Pro Rata Residence	1000	1000	1000	1000	1000
Depreciation	1000	1000	1000	1000	1000
Total Expenses	$3100	$3100	$3100	$3100	$3100
Profit (loss)	$(3100)	$(2100)	$(1100)	$ 900	$2900
Investment Tax Credit	$ 800				

qualifying your business

Now you are aware of the benefits. How do you get them? You must qualify your Lemonade business as "an activity engaged in for profit." You do this via the back door, that is, by not having the activity determined to be a hobby.

IRC Section 183 is commonly known as the "hobby loss". section and defines when an activity is not engaged in for profit. Ordinarily the IRC reads like a Faulkner novel; sentences go on for pages. However, IRC section 183 defines a "hobby" very tersely. There is no reference to "hobby" in the IRC. Section 183 defines "an activity not engaged in for profit" as "one other than an activity engaged in for profit." Such definition is typical IRC language and not much help.

However, Section 183 does provide a presumption that if the gross income

Detail of Aldo Giorgini's Silkscreen

☆ CHART 3

ADULT ENTERTAINMENT COMPUTER COMPANY
Expenses Which May Still Be Deducted
IRC Section 183 Presumption Not Met, Activity NOT Engaged in For Profit

	1977	1978	1979	1980	1981
Gross Income	$ 0	1000	2000	3000	6000
Less:					
deductions allowed without regard to whether activity engaged in for profit — interest $800	800	800	800	800	800
Income Remaining	(800)	200	1200	2200	5200
Less:					
other deductions which do not result in adjustment to basis rent, office expenses, trade journals $1300	0	200	1200	1300	1300
Income Remaining	(800)	0	0	900	3900
Less:					
deductions which result in adjustment to basis depreciation $1000	0	0	0	900	1000
Net Income	(800)	0	0	0	2900

derived from an activity in two or more taxable years in a period of five consecutive years exceeds the deductions attributable to such activity, such activity shall be presumed to be an activity engaged in for profit. In other words, if you can make money in two of five years, there is a presumption that you are engaged in an activity for profit.

To determine if you have qualified under the presumption, you count backwards using the present tax year as the fifth year. But, you say, that does me no good. The microcomputer industry is just over two years old. How can I possibly have four prior taxable years?

Take heart! The IRS allows you to elect to postpone determination with respect to the presumption until the end of the fifth year. If you so elect you may take the tax benefits described above as if you meet the presumption. Then, after the fifth year you look over your shoulder and see if, in fact, you did meet the presumption.

To so elect, you must file FORM 5212, ELECTION TO POSTPONE DETERMINATION and FORM 5214, CONSENT EXTENDING PERIOD OF LIMITATION. The reason for the first is self evident. The reason for the second is that ordinarily the Statute of Limitations for tax returns is three years. If you didn't meet the presumption, you could thumb your nose at the IRS for benefits obtained in the first two years. The IRS is smarter than to let that possibility arise.

Chart 2 demonstrates the principles of electing to postpone determination and the tax benefits available to the Lemonade entrepreneur. Chart 2 is actually the five year forecast which I have prepared for my own Lemonade operation, the Adult Entertainment Computer Company. I've decided to write an adult oriented higher level language. The mnemonic will be P.O.R.N.O. It stands for exactly what it spells. All commands will be obscenities.

In Chart 2 there appears an interest expense. I have assumed that I have purchased my $8000 computer using a 10% interest only loan (very unrealistic), of course.

It should be noted that the Lemonade entrepreneur may not wish to take an aggressive stance on his deductions, such as deducting a portion of residence or using accelerated depreciation. The reason is that by taking an aggressive position you might bootstrap yourself right out of the presumption that you are engaged in a business for profit. That is, your additional deductions may cause deductions to exceed income.

Also note that careful tax planning can help you to meet the presumption. For instance, wait until the end of the year to purchase that life subscription. If it appears that you are going to net $100 and the life subscription is $150, order the subscription in the following year instead.

If you fail to meet the presumption, you may still qualify your Lemonade business as an activity engaged in for profit. In such a case, there are a number of factors which are considered by the IRS in determining whether or not an activity is engaged in for profit. Among those factors are the following:

- The taxpayer's entire history of income or losses with respect to the Lemonade business.
- The amount of occasional profits, if any;
- The cause of the losses;
- The success of the taxpayer in carrying on other similar or dissimilar activities;
- The financial status of the taxpayer;
- The time and effort expended by the taxpayer in carrying on the activity;
- The manner in which the taxpayer carries on the activity;
- The expectation of profit by the taxpayer;
- The elements of personal pleasure or recreation.

The above factors are considered on a case by case basis, with no single one determinative.

Even if an activity is deemed to be one *not* engaged in for profit, there are some deductions allowed. Chart 3 illustrates these. The presumption was not met, because income in 1980 was only $3000 rather than $4000. Also assume the activity was not deemed to be an activity engaged in for profit under the factors previously mentioned.

First, deductions which are allowed without regard to whether or not an activity is engaged in for profit are subtracted from gross income. If there is any income left, other expenses may be deducted, but only to the extent that income is still left. Next are subtracted other expenses which do not result in adjustment to basis (generally nondepreciation deductions.) If income is still left, depreciation may be deducted but only to the extent of such income. If your activity is not engaged in for profit, your investment tax credit is not allowed.

Good luck! And remember that this approach doesn't cover everything. ■

GOLFCAP

by O.E. (Gene) Dial

In 1974, some 11 million golfers played over 280 million rounds of golf at over 12 thousand U.S. golf courses. In fact, managers at golf facilities worth almost four billion dollars spent five million dollars annually to maintain their courses and grounds. Golf is big business.

But all of these players and all of this activity create a problem: establishing and keeping current a relative index of each player's skill, permitting players of unequal ability to play together with equal opportunities to win. The solution to this problem is the handicap system whose success explains much of golf's popularity today.

In some ways it is a small miracle that the handicap system has survived the sheer mass of record keeping and volume of calculation required for its support. Let me illustrate in terms of averages constructed from the figures given at the outset.

If we distribute the 280 million rounds of golf over the 12,769 golf courses available, we find that 21,881 games were played on the average course. Further, if we distribute the number of rounds of golf among the 11,660,000 golfers, we find that the average golfer played 24 games. Now, if we assume that the average climate, given the probable distribution of golf courses, per-

mits nine months of play, and given the requirement that a new handicap must be posted for each player for each playing month, we are then ready to perform the extensions which tell us the magnitude of the task for each golf club. We find, in fact, that each course must calculate some 8199 handicaps per year. Now, **that's** a problem that you and your personal computer can solve.

Calculating a single handicap requires some 720 additions, 720 subtractions, a sorting routine, a division, a multiplication and an ordered listing. Our average club is therefore burdened with some 12 million calculations per year. But the fact is, this chore is divided between the golfer and the golf professional. And more recently, a new element has come to assist: the commercial computer.

A Natural for Lemonade

Today, the larger clubs contract with a computer service bureau to supply monthly handicap lists – a nuisance business to some bureaus. Frankly, it is a task for the Lemonade entrepreneur and his microcomputer. The business is more likely to be appreciated by both the amateur and the golf club. The simple, straightforward routine for calculating handicaps does require an understanding of a few golf terms. Further, a clear understanding must exist between the contracting parties about the division of responsibility in the production of handicap listings.

In amateur play, the players themselves score and audit scores. Experience indicates that this routine is a fairly reliable method, even though mistakes are occasionally made. These scores are submitted to the golf professional who records only the **gross adjusted score** for each player on a

(continued next page)

The Application of Mathematics to Golf

by Stephen Leacock

When Stephen Leacock wrote this classic essay some fifty years ago, the personal computer was not even a gleam in anyone's eye. Yet, the man's foresight was remarkable. With a few deft strokes of his pen, he outlined a field of activity that may preoccupy untold numbers of personal computing-minded golfers for years to come. Experts may cavil at the mathematics and logic presented by the author (a typical comment from critics has been "Rubbish!"), but the master plan of his work remains untarnished by time.

It is only quite recently that I have taken up golf. In fact I have played for only three or four years, and seldom more than ten games in a week, or at most four in a day. I have had a proper golf vest for only two years. I bought a "spoon" only this year and I am not going to get Scotch socks till next year.

In short, I am still a beginner. I have once, it is true, had the distinction "of making a hole in one," in other words, of hitting the ball into the pot, or can, or receptacle, in one shot. That is to say, after I had hit, a ball was found in the can, and my ball was not found. It is what we call circumstantial evidence – the same thing that people are hanged for.

Under such circumstances I should have little to teach anybody about golf. But it has occurred to me that from a certain angle my opinions may be of value. I at least bring to bear on the game all the resources of a trained mind and all the equipment of complete education.

In particular I may be able to help the ordinary golfer, or "goofer" – others prefer "gopher" – by showing him something of the application of mathematics to golf.

Many a player is perhaps needlessly discouraged by not being able to calculate properly the chances and probabilities of progress in the game. Take for example the simple problem of "going round in bogey." The ordinary average player, such as I am now becoming – something between a beginner and an expert – necessarily wonders to himself, "Shall I ever be able to go around in bogey; will the time ever come when I shall make not one hole in bogey, but all the holes?"

To this, according to my calculations, the answer is overwhelmingly "yes." The thing is a mere matter of time and patience.

Let me explain for the few people who never play golf (such as night watchmen, night clerks in hotels, night operators, and astronomers) that "bogey" is an imaginary player who does each hole at golf in the fewest strokes that a first-class player with ordinary luck ought to need for that hole.

Now an ordinary player finds it quite usual to do one hole out of the nine "in bogey" – as we golfers, or rather, "us goofers" call it; but he wonders whether it will ever be his fate to do all the nine holes of the course in bogey. To which we answer again with absolute assurance, he will.

The thing is a simple instance of what is called the mathematical theory of probability. If a player usually and generally makes one hole in bogey, or comes close to it, his chance of making any one particular hole in bogey is one in nine. Let us say, for easier calculation, that it is one in ten. When he makes it, his chance of doing the same with the next hole is also one in ten; therefore, taken from the start, his chance of making the two holes successively in bogey is one-tenth of a tenth chance. In other words, it is one in a hundred.

The reader sees already how encouraging the calculation is. Here is at last something definite about his progress. Let us carry it farther. His chance of making three holes in bogey one after the other will be one in 1000, his chance of four one in 10,000, and his chance of making the whole round in bogey will be exactly one in 1,000,000,000 – that is one in a billion games.

In other words, all he has to do is to keep right on. But for how long? he asks. How long will it take, playing the ordinary number of games in a month, to play a billion? Will it take several years? Yes, it will.

(continued next page)

** The Application of Mathematics to Golf appeared originally in a book called SHORT CIRCUITS, by Stephen Leacock, published by Dodd-Mead & Company, New York.*

Continued from p. 47.
monthly tally sheet presented to the service bureau at the end of each month.

Not yet accepted by The United States Golf Association (USGA), the term "gross adjusted score" buried in the preceeding paragraph is nonetheless useful. A gross score, ideally, would be one in which every stroke was recorded. The gross adjusted score would be the **gross score** reduced by those strokes a player was not permitted to record.

The USGA regards the gross score of a player to be the total number of strokes he required **and was permitted to record.** There is a limit to the number of strokes a player might record for a hole, and this limit is keyed to the player's handicap. Rules governing this adjustment were changed in 1976 to that which is set forth below:

- Golfers with handicaps from zero through 18 may record as many as two strokes over par on the number of holes equal to the handicap of the player.
- Golfers with handicaps from 19 through 36 may record as many as three strokes over par for as many holes as the handicap exceeds 18 strokes, but only two for those remaining.
- Golfers with handicaps from 37 through 40 are limited to four over par on as many holes as the handicap exceeds 36 and three over par for those holes remaining.

Adherence to this rule is the golfer's responsibility (as witnessed and audited by those with whom he is playing) in the same way that calculation of the total score is his responsibility. A service bureau is well advised **not** to include the verification of these portions of scoring in the service contract.

The total score recorded by the player is then the total number of strokes reduced by the ceiling imposed on each hole, consistent with the rules stated above. A further reduction, the handicap, is the equalizer between players, thus resulting in the **net score.**

The service bureau must avoid any responsibility for auditing. Each hole

Continued from p. 47.

An ordinary player plays about 100 games in a year, and will, therefore, play a billion games in exactly 10,000,000 years. That gives us precisely the time it will need for persons like the reader and myself to go round in bogey.

Even this calculation needs a little revision. We have to allow for the fact that in 10,000,000 years the shrinking of the earth's crust, the diminishing heat of the sun, and the general slackening down of the whole solar system, together with the passing of eclipses, comets, and showers of meteors, may put us off our game.

In fact, I doubt if we shall ever get around in bogey. Let us try something else. Here is a very interesting calculation in regard to "allowing for the wind."

I have noticed that a great many golf players of my own particular class are always preoccupied with the question of "allowing for the wind." My friend Amphibius Jones, for example, just before driving always murmurs something, as if in prayer, about "allowing for the wind." After driving he says with a sigh, "I didn't allow for the wind." In fact, all through my class there is a general feeling that our game is practically ruined by the wind. We ought really to play in the middle of the Desert of Sahara where there isn't any.

It occurred to me that it might be interesting to reduce to a formula the effect exercised by the resistance of the wind on a moving golf ball. For example, in our game of last Wednesday, Jones in his drive struck the ball with what, he assures me, was his full force, hitting it with absolute accuracy, as he himself admits, fair in the center, and he himself feeling, on his own assertion, absolutely fit, his eye being (a very necessary thing with Jones) absolutely "in," and he also having on his proper sweater – a further necessary condition of first-class play. Under all the favorable circumstances the ball advanced only 50 yards! It was evident at once that it was simply a matter of the wind: the wind, which was of that treacherous character that blows over the links had impinged full upon the ball, pressed it backward, and forced it to the earth.

Here, then, is a neat subject of calculation. Granted that Jones – as measured on a hitting machine the week the circus was here – can hit 2 tons, and that this whole force was pressed against a golfball only one inch and a quarter in diameter. What happens? My reader will remember that the superficial area of a golf ball is πr^3, that is 3.141567 X (5/8 inches)^3. And all of this driven forward with the power of 4000 pounds to the inch!

In short, taking Jones's statements at their face value, the ball would have traveled, had it not been for the wind, no less than 6½ miles.

I give next a calculation of even more acute current interest. It is in regard to "moving the head." How often is an admirable stroke at golf spoiled by moving the head! I have seen members of our golf club sit silent and glum all evening, murmuring from time to time, "I moved my head." When Jones and I play together I often hit the ball sideways into the vegetable garden from which no ball returns (they have one of these on every line; it is a Scottish invention). And whenever I do so Jones always says, "You moved your head." In return when *he* drives his ball away up into the air and down again ten yards in front of him, I always retaliate by saying, "You moved your head, old man."

In short, if absolute immobility of the head could be achieved, the major problem of golf would be solved.

Let us put the theory mathematically. The head, poised on the neck, has circumferential sweep or orbit of about 2 inches, not counting the rolling of the eyes. The circumferential sweep of a golf ball is based on a radius of 250 yards, or a circumference of about 1600 yards, which is very nearly equal to a mile. Inside this circumference is an area of 27,878,400 square feet, the whole of which is controlled by a tiny movement of the human neck. Inother words, if a player were to wiggle his neck even 1/190 of an inch the amount of ground on which the ball might falsely alight would be half a million square feet. If at the same time he multiplies the effect by rolling his eyes, the ball might alight anywhere.

I feel certain that after reading this any sensible player will keep his head still.

A further calculation remains – and one perhaps of even greater practical interest than the ones above.

Everybody who plays golf is well aware that on some days he plays better than on others. Question – How often does a man really play his game?

I take the case of Amphibius Jones. There are certain days, when he is, as he admits himself, *"put off his game"* by not having on his proper golf vest. On other days the light puts him off his game; at other times the dark; so, too, the heat; or again the cold. He is often put off his game because he has been up late the night before; or similarly because he has been to bed too early the night before; the barking of a dog always puts him off his game; so do children; or adults; or women. Bad news distrubs his game; so does good; so also does the absence of news.

All of this may be expressed mathematically by a very simple application of the theory of permutations and probability; let us say that there are altogether 50 forms of disturbance any one of which puts Jones off his game. Each one of these distrubances happens, say, once in ten days. What chance is there that a day will come when *not a single one of them occurs?* The formula is a little complicated, but mathematicians will recognize the answer at once as $x/1 + x^2/1 + \ldots x^n/1$. In fact, that is exactly how often Jones plays his best; $x/1 + x^2/1 \ldots x^n/1$ worked out in time and reckoning four to the week and allowing for leap years and solar eclipses, comes to about once in 2,930,000 years.

And from watching Jones play I think that this is about right.

on a golf course is rated not only with respect to par, but also in terms of relative difficulty in making par with respect to the remaining holes. This means, for example, that if a player has a handicap of "1", he may apply that handicap only with respect to the hole on the course which is judged to be the most difficult to make par. Now the key point, at least from your perspective, is that the distribution of par-ratings for each hole is unique to each golf course. The most difficult hole to make par on one course may be the second, and on another the fifteenth, etc.

It should be evident that a service bureau which includes the auditing of player scores at the hole-level must multiply data inputs by a factor of 18, and program length by a factor of about five. The service can be provided, but the golf club must be prepared to pay.

The Golfcap Program

The BASIC program provided with this article makes a number of key assumptions. The reader will need to revise the program where the operating environment differs from those described in the assumptions. They are:

o Operating facilities include a microcomputer with at least 12K RAM, a BASIC interpreter, a CRT with keyboard and a printer.

o The golf club will supply its course rating and a monthly list of players and their adjusted gross scores for games during the period; and the list may be retained by the service bureau.

o The service bureau is responsible to return a list displaying the name of each golfer and his current handicap. Also shown will be the name of the course and the date of the assigned handicaps.

Note that most data is entered in data statements in the program rather than as input statements to permit easier editing and revision. The method also assures that the input data will be recorded with the program, e.g., on cassette, paper tape or disk for purposes of storage until it is next used. At that time only the new data need be entered and a portion of the old data erased. Note that much of the old data will continue to be needed in the new run.

Also note that matrices are minimized in the program to reduce the level of memory required. All processing of one player's scores is completed before beginning the processing of scores of the next player. Should the limits of memory become a problem, there are more opportunities for economizing in the program. Much of it can be realized by statement consolidation, the elimination of remarks and all of the operating instructions. Beyond that, adjust the CLEAR statement to that which is actually required for the particular run. (See statement #50). The deletion of all CONSOLE instructions will be required if the program is to operate in a printer-only environment.

On the other hand, the program could be usefully improved by adding a routine which compares the new handicap with the old and prints out a flag (i.e., an asterisk beside the handicap listing) where there has been a change. Some service bureaus currently do this. Given the assumptions of the operating environment for this program, however, this feature has been omitted.

The handicap algorithm (statement #402) is a simple one, requiring only that the course rating be subtracted from the average of the best ten of the player's most recent twenty scores. The result, as of last year, is then multiplied by 95%, giving you the player's handicap.

Since in many instances there will be fewer than 20 scores, and in some, fewer than 10 scores, the program must be prepared to meet this contingency thus adding its only complication. This is done by requiring that a zero be listed following the last score in each instance of fewer than 20 scores.

Now assuming that this proves to be an assist to those of you who have Lemonade Enterprises – the marketing idea and the program – you are now obliged to straighten out my program's slice (and hook, too).

```
                    * GOLFCAP RUNNING INSTRUCTIONS *

STATEMENTS FROM 1000 AND UPWARD ARE RESERVED FOR YOUR DATA STATE-
MENTS.  THESE ARE USED TO SET FORTH THE ROSTER OF PLAYERS, THEIR
TWENTY MOST RECENT SCORES, THE NAME OF THE COURSE, AND ITS PGA RATING.

STATEMENT NUMBER 1000 SHOULD BE USED FOR THE NAME OF THE COURSE,
IN QUOTES, FOLLOWED BY A COMMA, THEN THE PGA COURSE RATING.

THE REMAINING STATEMENTS SHOULD BE TREATED IN PAIRS.

   1. THE FIRST STATEMENT IN EACH PAIR SHOULD SET FORTH THE PLAY-
      ER'S NAME, IN QUOTES, AND LAST NAME FIRST.

   2. THE SECOND STATEMENT IN EACH PAIR SHOULD LIST THE TWENTY MOST
      RECENT SCORES OF THE PLAYER, EACH SCORE SEPARATED BY A COMMA.
      IF A PLAYER HAS FEWER THAN TWENTY SCORES, THEN A ZERO SHOULD
      BE ENTERED FOLLOWING HIS LAST SCORE.

THE REMAINING VARIABLE, I.E., THE DATE, WILL BE ASKED FOR AS AN
INPUT STATEMENT.

THE USER NEED NOT PRE-SORT THE NAMES OF THE PLAYERS.  THE FUNCTION
OF THE PROGRAM IS TO PRODUCE AN ALPHABETIZED LIST OF PLAYERS AND
THE HANDICAP OF EACH.

YOU ARE ABOUT TO BE ASKED HOW MANY PLAYERS ARE LISTED ON YOUR ROS-
TER.  ALL THAT IS NEEDED IS A ROUGH ESTIMATE, BUT THE ESTIMATE MUST
BE GREATER THAN THE NUMBER OF PLAYERS.  IF YOU DO NOT HAVE A COUNT,
DIVIDE THE NUMBER OF DATA STATEMENTS YOU HAVE USED BY TWO AND ADD TEN.
```

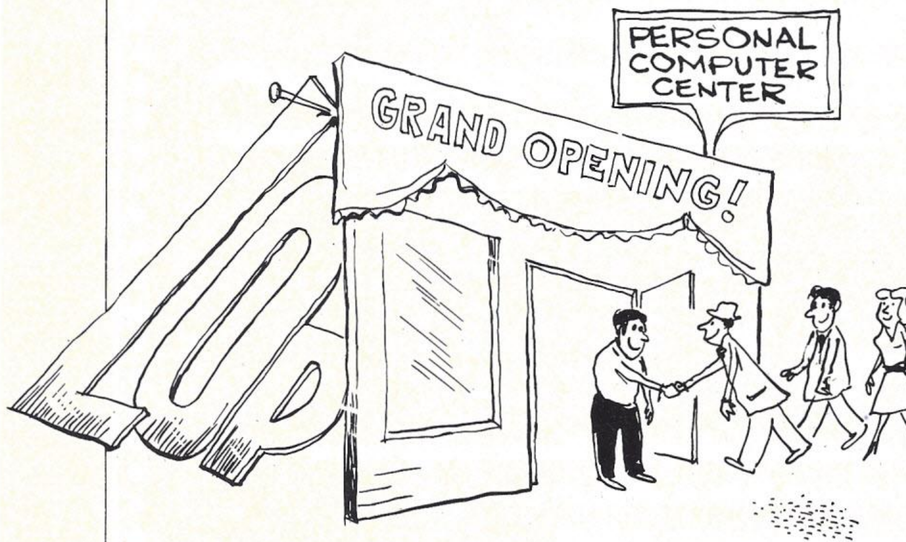
```
HOW MANY PLAYERS ON THE ROSTER? 100
WHAT IS THE DATE (PLEASE USE NO COMMAS)? JANUARY 1977
PAPER POSITIONED?
```

```
GOLFCAP CODE LIST     O. E. DIAL
LIST
50 CLEAR 1000: PRINT TAB(23)" * GOLFCAP *":PRINT
60 INPUT"DO YOU DESIRE RUNNING INSTRUCTIONS (Y OR N)";X$
70 PRINT:IF X$<>"Y" THEN 340 ELSE CONSOLE 18,8:  PRINT TAB(19)"* GOLFCAP RUNNING INSTRUCTIONS *"
79 '
100 PRINT: PRINT"STATEMENTS FROM 1000 AND UPWARD ARE RESERVED FOR YOUR DATA STATE-"
110 PRINT"MENTS.  THESE ARE USED TO SET FORTH THE ROSTER OF PLAYERS, THEIR"
120 PRINT"TWENTY MOST RECENT SCORES, THE NAME OF THE COURSE, AND ITS PGA RATING.":PRINT
129 '
130 PRINT"STATEMENT NUMBER 1000 SHOULD BE USED FOR THE NAME OF THE COURSE,"
140 PRINT"IN QUOTES, FOLLOWED BY A COMMA, THEN THE PGA COURSE RATING.":PRINT
149 '
150 PRINT"THE REMAINING STATEMENTS SHOULD BE TREATED IN PAIRS.":PRINT
160 PRINTTAB(3)"1. THE FIRST STATEMENT IN EACH PAIR SHOULD SET FORTH THE PLAY-"
170 PRINTTAB(6)"ER'S NAME, IN QUOTES, AND LAST NAME FIRST.":PRINT
180 PRINTTAB(3)"2. THE SECOND STATEMENT IN EACH PAIR SHOULD LIST THE TWENTY MOST"
190 PRINTTAB(6)"RECENT SCORES OF THE PLAYER, EACH SCORE SEPARATED BY A COMMA."
200 PRINTTAB(6)"IF A PLAYER HAS FEWER THAN TWENTY SCORES, THEN A ZERO SHOULD"
210 PRINTTAB(6)"BE ENTERED FOLLOWING HIS LAST SCORE.":PRINT
219 '
220 PRINT"THE REMAINING VARIABLE, I.E., THE DATE, WILL BE ASKED FOR AS AN"
230 PRINT"INPUT STATEMENT.":PRINT:PRINT
239 '
240 PRINT"THE USER NEED NOT PRE-SORT THE NAMES OF THE PLAYERS.  THE FUNCTION"
250 PRINT"OF THE PROGRAM IS TO PRODUCE AN ALPHABETIZED LIST OF PLAYERS AND"
260 PRINT"THE HANDICAP OF EACH.":PRINT
269 '
300 PRINT"YOU ARE ABOUT TO BE ASKED HOW MANY PLAYERS ARE LISTED ON YOUR ROS-"
320 PRINT"TER.  ALL THAT IS NEEDED IS A ROUGH ESTIMATE, BUT THE ESTIMATE MUST"
330 PRINT"BE GREATER THAN THE NUMBER OF PLAYERS.  IF YOU DO NOT HAVE A COUNT,"
332 PRINT"DIVIDE THE NUMBER OF DATA STATEMENTS YOU HAVE USED BY TWO AND ADD TEN.":PRINT
333 '
334 CONSOLE 16,8
335 '
340 PRINT:INPUT"HOW MANY PLAYERS ON THE ROSTER";NP: PRINT
349 '
350 DIM PL$(NP), H(NP), PS(19)
360 READ CN$,CR:INPUT"WHAT IS THE DATE (PLEASE USE NO COMMAS)";D$
361 '
362 PRINT:PRINTTAB(21)"* NOW WORKING *":PRINT
367 '
368 '                    THE READ LOOP
369 '
370 FOR J=0 TO NP
372     READ PL$(J)
374     IF PL$(J)="DONE" THEN NP=J-1:GOTO430'   TO ESCAPE THE LOOP WITH THE EXACT NUMBER
375 '                                           OF GOLFERS (NP) WHEN ALL DATA IS READ.
376     FOR N=0 TO 19
377             READ PS(N): IF PS(N)=0 THEN N=N-2: GOTO 382
378     NEXT N: N=N-1
379 '
380 '                    THE SORT LOOP
381 '
382     FOR K=0 TO N: F=0'                      NOW WE STACK THE ARRAY FROM LOW TO HIGH SCORES.
383 '                                           THE SWAP FLAG 'F' IS USED SO IT CAN BE KNOWN WHEN
384 '                                           THE ENTIRE LOOP WAS CYCLED WITHOUT A SWAP.
385             FOR L=0 TO N-1
386                     IF PS(L)<=PS(L+1) THEN 389
388                     SWAP PS(L), PS(L+1): F=1'       TO SET THE SWAP FLAG.
389             NEXT L: IF F=0 THEN 394'          'F' WILL BE ZERO IF NO SWAPS OCCURED AND THE DATA
390     NEXT K'                                 WILL BE ARRAYED.
391 '
392 '                    THE CALCULATIONS LOOP
393 '
394     FOR K=0 TO 9
396             IF PS(K)=0 THEN A=K: GOTO402'   THERE MAY BE FEWER THAN 10 SCORES FOR SOME GOLFERS,
397 '                                           HENCE THE DIVISOR 'A' IS DEFINED.
398             W=W+PS(K)
400     NEXT K: A=10
402     H(J)=W/A: H(J)=.95*(H(J)-CR): W=0'      THE HANDICAP ALGORITHM, THEN 'W' IS INITIALIZED.
404 NEXT J
426 FOR J=0TO NP: PRINT PL$(J), H(J): NEXT J
427 '
428 '                    THE ALPHABETICAL SORT LOOP
429 '
430 FOR J=0 TO NP: F=0 '                        'F' IS A SWAP FLAG.
440     FOR K=0 TO NP-1
450             IF PL$(K)<=PL$(K+1) THEN 480
460             SWAP PL$(K), PL$(K+1)
470             SWAP H(K), H(K+1): F=1'         'F' WILL BE SET TO '1' IF A SWAP OCCURS.
```

```
480     NEXT K: IF F=0 THEN 500'                    NO SWAP OCCURED, SO THE SORT MUST BE FINISHED.
490 NEXT J
497 '
498 '                    THE PRINT ROUTINE
499 '
500 INPUT"PAPER POSITIONED";X$:W1$="###":W2$="###. ":CONSOLE 18,8
510 PRINTTAB(18-(LEN(CN$)/2))CN$:PRINT: PRINTTAB(18-(LEN(D$)/2))D$:PRINT
520 FOR J=0 TO NP
530     PRINTUSING W2$; J+1;'           PRINT USINGS ARE USED TO RIGHT JUSTIFY NUMERICAL COLUMNS,
531 '                                   AND TO SUPPLY OTHER SYMBOLS OR SPACES, E.G., SEE W2$.
532     PRINT PL$(J)" ";
540     FOR K=0TO 25-LEN(PL$(J)):PRINT".";:NEXTK:' TO PRODUCE ELLIPSES BETWEEN NAME AND HANDICAP.
570     PRINTUSINGW1$;H(J)
580     IF INT((J+1)/3)=(J+1)/3 THEN PRINT'      TO PRODUCE A  BLANK LINE EACH 4TH LINE
590 NEXT J
599 '
600 CONSOLE16,8:STOP
997 '
998 '                    TEST DATA
999 '
1000 DATABOULDER COUNTRY CLUB,71.3
1001 DATA"HANDEL, JOHN"
1002 DATA100,90,80,99,89,102,103,102,100,85,0
1003 DATA"FRAMINGHEL, BILL D."
1004 DATA99,90,89,79,88,0
1005 DATA "TROYER, PAUL"
1006 DATA.100,102,104,105,101,99,110,104,109,102,103,100,101,100,105,110,102,103,101,110
1007 DATA "HICKMAN, BILL"
1008 DATA78,76,80,83,82,81,79,84,87,84,86,87,89,89,89,90,91,87,88,87
1009 DATA"WUNSCH, ED"
1010 DATA79,76,78,75,80,81,86,85,88,89,83,84,79,78,76,75,75,79,80,78
1999 '
2000 DATADONE
```

```
                    BOULDER COUNTRY CLUB     APRIL 1977

  1. FRAMINGHEL, BILL D. ........ 17     4. TROYER, PAUL .............. 28
  2. HANDEL, JOHN ............... 23     5. WUNSCH, ED ................  5
  3. HICKMAN, BILL .............. 10
```



Part I
Retail Computer Stores: Before You Open The Doors
LOOK OUT FOR LOP

Computer retailing is a new, disorderly and exciting field with almost hypnotic appeal to adventuresome people who have been hooked on personal computing. Whether you're eager to become a retailer yourself or get involved with retailers as a plain old customer, these words of wisdom from a professional consultant in the field may be both entertaining and useful.

By Paul Conover
Consumer Computer Marketing, Inc.

Perhaps 200 retail computer stores have opened all over the country in the last year and a half. Many of these are doing very well, operated by ambitious entrepreneurs with some computer-related experience who managed to attract partners or employees harmonious with themselves. By pure luck, in almost all cases, they have created a synergy that works.

Most entrepreneurs have not enjoyed this good luck. They have a hard time making ends meet. Three-quarters of these struggling stores have something in common: they were started by non-business-oriented people who really didn't have a detailed plan of what they were doing.

You can't fault them for trying; it's the American Way. The established small-business philosophy of hard work and an honest dollar are almost universally attractive to the average guy with a little cash to invest in himself. Making your money work for you is fine, but *participating* with your money is better. Every day people decide to go into business for themselves.

New and changing markets are always opportunities for profit. Just ask Horatio Alger, Jeno Paulucci or William Lear. Everyman can succeed, even today. Look what's happening with micro-computer/retail computer outlets. Here is an emerging opportunity. The demand/supply ratio seems very healthy. Experts in the computer field are hot and bothered over the applications of micros and their potential market; everything that has to do with microcomputers is lookin' rosy as hell. More customers are coming out of the woodwork than even the most ebullient editorials of the microcomputer journals can describe. It looks to this writer like a bona fidé chance for Everyman to "make it".

Head on the Block

My advice to anyone going into any business, including computer stores, is "know what you're doing".

Last year, while talking with dozens of prospective Altair Computer Center owners I encountered several such opportunity-seekers. One, a wig-shop owner in a large Midwestern city, was determined to open a computer store one way or the other. He declined the MITS Altair Computer Center offer because he "didn't like to do paper work"; the deal called for him to make summary reports of his usual monthly accounting records. Another very anxious individual in the Southwest was unwilling, after long talks, to pay a "fee" for learning to run a business that he has already carefully investigated. So I suggested that he check into the IMSAI Dealer Program; they didn't have strict requirements (at that time). His careful research and considered examination of the microcomputer market provided him with a snappy come-back to my suggestion: "Who's IMSAI?" Still others would pursue long technical discussions of the merits of this memory card over that one. Very, very few had pertinent questions about the business itself.

There is a common denominator here. The problem encountered by the current store owners and the would-be store owners is the same for most beginning businessmen: Lack Of Planning. We'll call this the LOP factor. In small business operations you can get your head LOPped off if you don't know what you're doing or going to do.

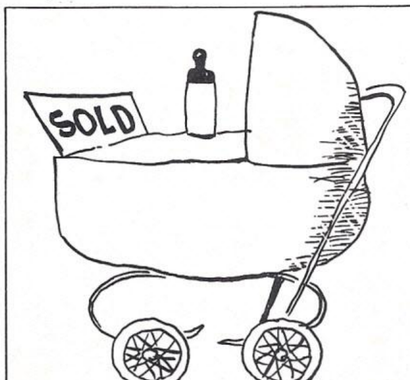
The largest bank on the West Coast published a great series of pamphlets on small business, its funding and operations. One pointed out that nine out of every ten new businesses failed in their first two years for the same reasons: lack of capital and poor management. Poor business management, starting long before the doors opened, plus insufficient money to keep the business afloat over the inevitable down spots in any business operation killed these fledgling enterprises. With proper planning, the capital shortage wouldn't have been as likely.

The same series of pamphlets raised another point. Franchises had a much better success rate. Only one out of eleven franchisees/franchisors failed. Great! There must be significant advantages to the entreprenuer who selects the franchise route. Even franchising doesn't preclude the absolute necessity for a well-documented pre-business plan. Somehow the average guy must plug into tried and true techniques that will inform him completely so that he can make intelligent decisions.

Ask the Questions

In retail computer stores, both the franchisors (there are three now) and the independents have some track records. Before hocking your first-born child, start learning how to gather the information you need. You've got to know about business operations and the complexities, physical demands and financial commitments and planning essential to success in any business.

How do you do that? Well, thoroughly. Let's make a list, a program for business. Start with the entrepenuer. You. *What personal qualities do you have which will help you succeed in your new business? Physical energy? Leadership experience? Organizing ability? Determination and perseverence?* These are all necessary, in spirit-breaking amounts, to the success of any business venture. *What have you done to prepare youself for business ownership? Do you have previous management experience in a successful small business? Taken classes? Personal savings? Extensive use of several public libraries? Business contact? What makes you so sure you're ready to do all of this?* The list you're making had better run a couple of closely spaced pages up to this point. Make a list of all the reasons



...BEFORE HOCKING YOUR FIRSTBORN CHILD...

that you want your own business and then talk with advisors. Friends, relatives, devil's-advocates, clergy, bankers, lawyers and Indian Chiefs all can give you insight into the validity of your reasons. This applies to any new business, computer stores or campgrounds.

Now focus on computer stores. *Have you done your homework on computers? Suppliers? Prices? Retail computer*

... HIS OWN BOSS...

stores? What are trends? You'd better consult several sources on that one. *What will happen to computer stores if something entirely new comes into the microcomputer field? When is that likely to happen? Is it likely at all? Have you talked with several different franchise operations? Independent store owners?* (More on that later). *What makes computer stores such a hot idea? Who says so? Why is it hot for you?* So very many of the people who are interested in opening businesses have never run a business before!

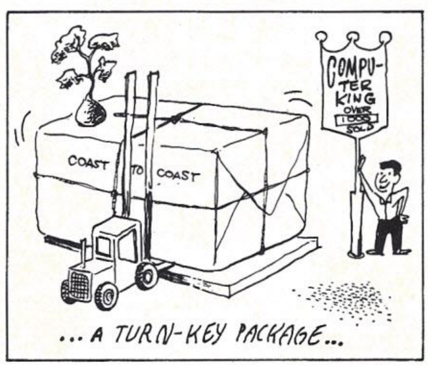
What about financing the business? Do you have a complete itemized list of all the funding needs for starting this business? How about operating expenses for the first year or two? Despite the rosiest projections or the experiences of someone else in another part of the country, if that extra money is needed, and you don't have it, you might see your investment, mortgage and first-born child sold to the highest bidder. *Who's going to pay for your normal living expenses, taxes and vacations while you start-up a business?* Whoops. Scratch vacations. You're going into business. Forget about vacations for a long while. Onward. *What provisions do you have for extra cap-*

ital if needed? What independent expertise have you sought to support your capitalization outline?

If you plunge into business, learning as you go, you're a pigeon for the LOP factor. Some more important considerations: *Business policies? How will you allocate your time? Employees and wage scales? Incentives? How much insurance on you, your employees, the business? Outside suppliers? Increased competition? Sales promotions? Who are your customers, really? Customer relations? Local advertising media? Which one works best for computer stores? How do you know? What guideline will you measure the effectiveness of your advertising and promotions with? What about credit customers? How do you check them out? What controls do you have on credit sales? What is a balanced inventory? Inventory control? What system? What is your breakeven volume? How will you control your expenses? How many alternative budgets do you have? What are the standard operating ratios for computer stores? How about compliance with local, county, state and Federal regulations? Provisions for taxes? What record keeping system will comply with all of this and still work for you, not you for it?* Each of these questions deserves its own section.

Help is Available

In the legwork process of getting "into business for yourself" you'll need advice and services of four specific professionals. Right from the beginning. Talk over what you're thinking about (say, opening a computer store) with your lawyer, accountant, banker and insurance broker. If you don't already have these people lined up, here's more legwork for you. You can certainly call a few, outlining your need for their



...LONG BEFORE THE DOORS OPENED...

type of services or expertise. Be sure to explain that you're searching for the right man. Most professional accountants or lawyers will understand that this initial visit is for "feeling out" the prospects of a good working relationship and will allocate time to attract a new client without charging you for the visit. If you start talking specifics with them though, you may be requested to pay for the visit.

Your attorney can advise you on the legal ramifications of every aspect of any agreement you may be considering. The business decision is yours to make, however. An accountant can be very helpful in the evaluation of true costs and future expenses in the operation of the business. His experience with similar start-ups can yield very significant savings and financial structures that will steer clear of money shortages. The business decisions are yours again, however.

Your banker can be your best ally. He's in the business of renting money, and will either help with your business planning or suggest someone else in the bank who can help. If he doesn't have the experience or the interest in helping you with your investigation and business plan, shop around for another bank that does more commercial business. It might even be helpful for you to talk with other computer store owners about the problems they've had in finding a banker knowledgeable in microcomputers. You might even try to use the same bank as one used by a computer store in your area.

A professional insurance man should also be on the planning team. There are over thirty policy areas that need consideration by a new business venture. A brokerage firm that has been involved in a number of new business start-ups can balance the insurance needs of a new venture against the cash flow parameters they're used to seeing.

Even with this crew of experts on your side, the ultimate business decision: go or no-go, rests with you. Have you done your homework so far? Are you satisfied that you understand enough about the retail computer business to be able to evaluate the opportunities with the franchise operations? Consider franchising.

Why is franchising so attractive? It's successful. Ideally, the franchise system of product distribution offers many benefits to both the franchisor and the franchisee. Without heavy investments in time or money the franchisor can expand or start-up a network of outlets that sell its product and services. Great for the franchisor. The franchisee receives the benefits of running his own business, psychologically being "his own boss". He profits directly from the fruits of his own money and labors. He receives advantages usually reserved for Big Business – national and regional advertising, mass purchases of goods at reduced prices (generally), and local identification with a well-known (hopefully) product or service..

The franchising system is designed to overcome the major failure of new businesses: poor management and planning. It gives the franchisee a "turn-key" package that is tested and works.

Whenever franchising is talked about the images of the golden arches, Holiday Inns or that finger-lickin'-chicken appear before your eyes. Franchise sales should account for 31% of all retail sales in the U.S. in 1977. The entreprenuer who searches diligently for a small, not widely known franchise operation in hopes that this opportunity can be ridden to the top may be well rewarded.

Before the fledgeling businessperson can coherently discuss a business proposition with the various franchisors, each item on the personal pre-business check list should be filled in. *Then* it's time to talk with the franchise guys.

In Part II of this series, we'll look closely at franchising as an uncertain guard against LOP.

This brochure explains and demonstrates the elements of the POLY 88 system, a few of its applications, and shows an actual "run-through" of a specialized program. Please read through the brochure and take it along when you visit your computer dealer to operate the POLY 88 for yourself.

POLY 88 DEALERS

ALABAMA
Computer Center, 303B Poplar Place, Birmingham, AL 35209

ARIZONA
Byte Shop Arizona, 813 N. Scottsdale Rd., Tempe, AZ 95282
Byte Shop Phoenix, 12654 W. 28th Dr., Phoenix, AZ 95209
Byte Shop Tucson, 2612 E. Broadway, Tucson, AZ 85716

CALIFORNIA
Algorithm Personal Computers, 7561 Rhine Dr., Hunt. Beach, CA 92647
Bits N Bytes, 679 D S. State College Blvd., Fullerton, CA 92631
Byte Shop Berkeley, 1514 University Ave., Berkeley, CA 94703
Byte Shop Burbank, 1812 Burbank Blvd., Burbank, CA 91506
Byte Shop Lawndale, 16508 Hawthorne Blvd., Lawndale, CA 92060
Byte Shop Mountain View, 1063 W. El Camino Real, Mt. View, CA 94040
Byte Shop Palo Alto, 2227 El Camino Real, Palo Alto, CA 94306
Byte Shop San Diego, 8250 Vickers Rd., San Diego, CA 92111
Byte Shop San Fernando, 18424 Ventura Blvd., Tarzana, CA 93156
Byte Shop San Jose, 2626 Union Ave., San Jose, CA 95124
Byte Shop Santa Barbara, 4 W. Mission St., Santa Barbara, CA 93103
Byte Shop Santa Clara, 3400 El Camino Real, Santa Clara, CA 95051
Byte Shop Thousand Oaks, 2705 T. O. Blvd., Thousand Oaks, CA 93160
Byte Shop Ventura, 2409 E. Main, Ventura, CA 93003
Computer Center, 1913 Harbor Blvd., Costa Mesa, CA 93626
Computer Components, 5848 Sepulveda Blvd., Van Nuys, CA 91406
ComputerLand, 22634 Foothill Blvd., Hayward, CA 94542
ComputerLand, 6840 La Cienega Blvd., Inglewood, CA 92302
ComputerLand, 16919A Hawthorne Blvd., Lawndale, CA 92060
ComputerLand, 103 W. First St., Tustin, CA 92680
Computer Mart, 627 W. Katella, Orange, CA 92667
Computer Room San Jose, 155 Blossom Hill Rd., San Jose, CA 95123
Computer Terminal, 309 S. San Mateo Dr., San Mateo, CA 94401
Computer World Store, 1309 Court St., Redding, CA 96601
Jef Raskin, 275 Humboldt Rd., Brisbane, CA 94005
MicroComputers, 18120 Brookmist, Fountain Valley, CA 92708
OPAMP/Computer, 1033 N. Sycamore Ave., L. A., CA 90038
SP Distributors, 1477 Barrington, Suite 17, Los Angeles, CA 90025
Upland Computer Labs, 75 E. Ninth St., Upland, CA 91799
XI Media, 1290 24th Ave., San Francisco, CA 94122

COLORADO
Byte Shop Arapahoe County, 3464 S. Acoma, Englewood, CO 80110
Byte Shop Boulder, 2040 30th St., Boulder, CO 80301

CONNECTICUT
National Communications Inc., 50 Washington St., Norwalk, CT 06850

FLORIDA
Computer Hut, 5905 Northwest 151st St., Miami Lakes, FL 33014
Economy Computing Systems, 2200 Forsyth Rd., Orlando, FL 32807
Personal Computer, Ft. Lauderdale, FL
Solartronics, 2506 S. Hopkins, Titusville, FL 32952

GEORGIA
Atlanta Computer Mart, 5091-B Buford Hwy., Atlanta, GA 30304

HAWAII
Capacity Inc., P.O. Box A, Haiku, Maui, HA 96708
MicroComputer Systems of Hawaii, Kukui Plaza, Honolulu, HA 96813

ILLINOIS
Aspen Computer, 7519 W. Irving Pk. Rd., Chicago, IL 60634
Essence Electronics, 169 North Merion, Oak Park, IL 60301
ittybitty machine company, 1316 Chicago Ave., Evanston, IL 90204
ittybitty machine company, 42 W. Roosevelt, Lombard, IL 60148
Numbers Racket, 518 East Green, Champaign, IL 61820

INDIANA
Byte Shop Indiannapolis, 2947 East 82nd St., Indiannapolis, IN 46250
Home Computer Center, 10447 Chris Dr., Indiannapolis, IN 46229
Valpariaso Technical Inst., 1 Center St. Valpariaso Technical Campus Valparaiso, IN 46383

IOWA
Micro Bus Inc., 1910 Mt. Vernon Rd. SE., Cedar Rapids, IA 52403

KANSAS
Computer Systems Design, 1611 E. Central, Wichita, KS 67214

KENTUCKY
ComputerLand of Louisville, 813 B Lyndon Ln., Louisville, KY 40222

LOUISIANA
Computer Shoppe Inc., 344 Camp St., New Orleans, LA 80130

MARYLAND
Computer Workshop, 1776 Plaza, 1776 E. Jefferson, Rockville, MD 20852

MASSACHUSETTS
Computer Mart, Inc., 1097 Lexington St., Waltham, MASS 02154
Computer Shop, 288 Norfolk St., Cambridge, MASS 02139

MICHIGAN
CompuMart Inc., 1250 N. Main St., Ann Arbor, MI 48104
Computer Systems, 26401 Harper, St. Clair Shore, MI 48081
Data Mart Inc., 17117 W. 9 Mile Rd., Southfield, MI 48075
General Computer Store, 2011 Livernois, Troy, MI 48084

MINNESOTA
Computer Depot, Inc., 3515 W. 70th St., Edina, MINN 55435

MISSOURI
Advanced Data Management, Rm 411 Plaza Tower, 176 E. Sunshine, Springfield, MO 65084
Computer Workshop Kansas City, 6903 Blair Rd., Kansas City, MO 64152
H & K Company, 15 E 31st St., Kansas City, MO 64108
Micro-Com Inc., 6314 Brookside Plaza, Suite 202, Kansas City, MO 64113

NEBRASKA
Omaha Computer Store, 4540 S. 84th St., Omaha, NB 68127

NEW HAMPSHIRE
MicroComputers Inc., 539 Amherst St., Nashua, NH 03036

NEW JERSEY
ComputerLand of Morristown, 2527 Andy De-Hart St., Morristown, NJ 07960
Hoboken Computer Works, 20 Hudson Place, Hobkewn, NJ 03887

NEW YORK
Computer Corner, White Plains Mall, 200 Hamilton Ave., White Plains, NY 10601
Computer Enterprises, 3307 Erie Blvd., Syracuse, NY 13214
Computer Mart of New York, 118 Madison Ave., New York, NY 10016
Computer Mart of New York, 118 Madison Ave., NY, NY 10016
Computer MicroSystems, 1311 Northern Blvd., Manhassett, NY 11030

NORTH CAROLINA
Systems Management Corp. One NCNB Plaza, Charlotte, NC 28231

OREGON
MicroMethods, P.O. Box 143, 118 SW 1st, Warrenton, OR 97246

OHIO
CyberShop, 1451 S. Hamilton Rd., Columbus, OH 53227

PENNSYLVANIA
2005 AD, 2005 Naudain St., Philadephia, PA 10145

SOUTH CAROLINA
Byte Shop Columbia, 2018 Green St., Columbia, SC 29250
World of Computers, 5849 Dorchester Rd., Charleston, SC 29405

TEXAS
Byte Shop Computer Store, 3211 Frondren SE, Houston, TX 77063
CompuShop, 13933 Frondren SE, Houston, TX 77063
Computer Shop, 6812 San Pedro, San Antonio, TX 78216
Computer Terminal, 2101 Myrtle, El Paso, TX 79991
Micro Store, 634 S. Central Expressway, Richardson, TX 75080

UTAH
Computer Room, Inc., 3455 South West Temple, Salt Lake City, UT 84115
Computers & Stuff, 1717 W. Center St., Provo, UT 84601
MicroData Systems, 796 E. Lazon Dr., Sandy, UT 84070

VIRGINIA
Computer Hobbies, 9601 Kendrick Rd., Richmond, VA 23235
Computer Systems Store, 1984 Old Chain Bridge Rd., McLean, VA 22101

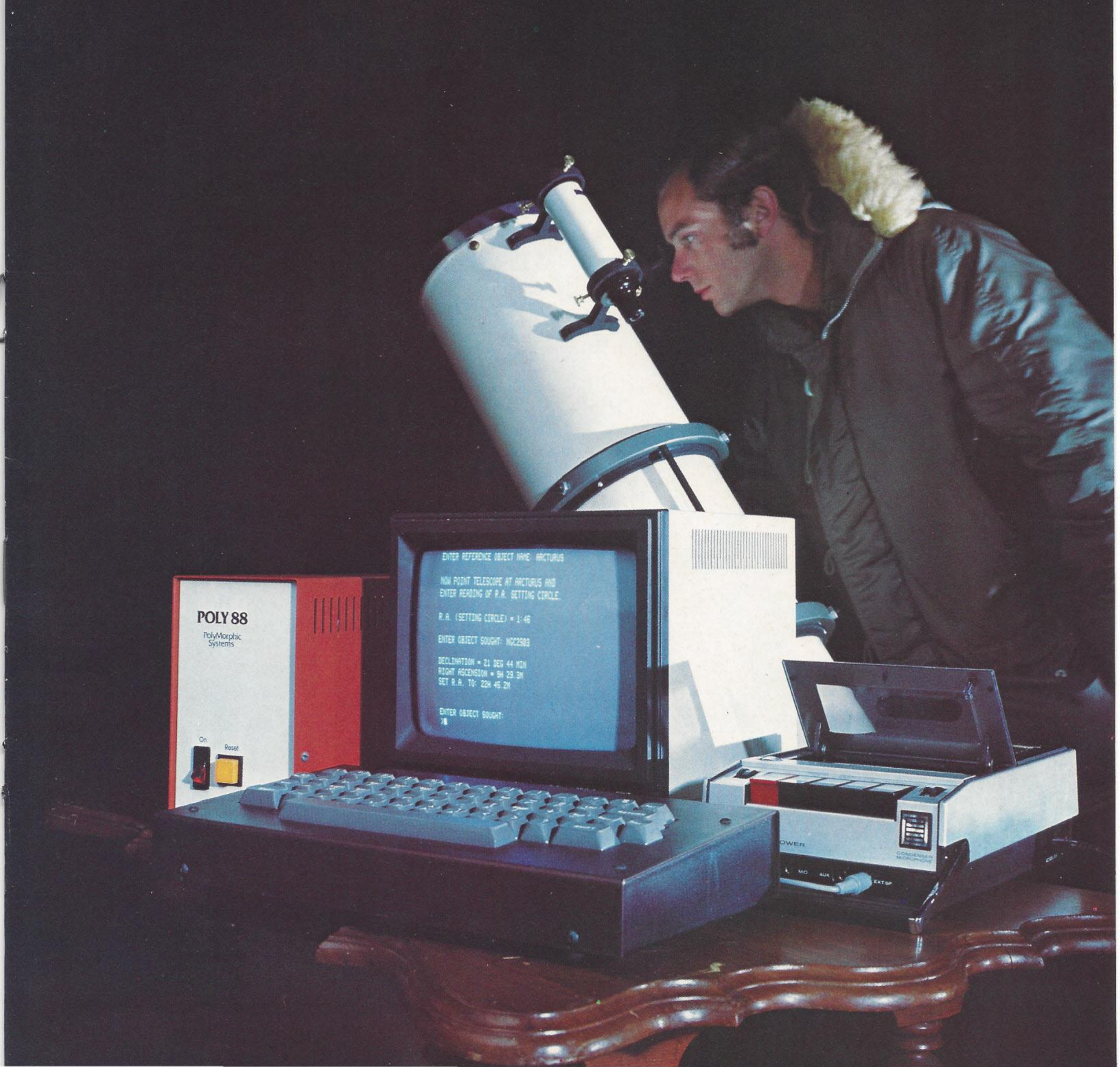
WISCONSIN
Madison Computer Store, 1910 Monroe St., Madison, WI 53711

DEALERS IN CANADA
Computer Shop, 3515 18th St. SW, Calgary, Alberta T2T 4T9
Computer Mart Ltd., 1543 Bayview Ave., Toronto, Canada M1K 4K4
Dynapro Systems, Inc., 875 West Broadway, Vancouver, BC Canada
Microbyte Computers, Inc., 250 West Broadway, Vancouver, BC Canada

DEALERS OVERSEAS
Byte Shop Sogoh, Towa Bldg. 1-5-9 Sotodanda Chiyodaku, Tokyo, Japan
Compelec Electronics, 310 Kilburn High Rd., London, England NW6
Digitronics, Bei der Doppeleiche, 2000 Wedel (Holstein), Germany
Dynetics Pty Ltd., 425 E. Pennant Hills Rd., Pennant Hills, NSW 2120 Australia

If insert is missing, CIRCLE 23 ▶

The POLY 88 Microcomputer System



If someone offered to sell you a computer to use in your home, you'd probably think he was nuts.

What on earth would you use it for? Where would you have room to put it? How could you afford to supply it with power, programs and peripheral equipment? And who would operate it for you?

The answers are suprisingly simple and practical. They come in the form of an extraordinary and unique personal tool, the POLY 88 microcomputer by PolyMorphic Systems.

Your entire family will be able to use and enjoy the POLY 88. It is a perfect introduction for children to the world of computers, stimulating them to use their intelligence and imagination. You can easily learn to live with the POLY 88—in fact it may become a highly valued member of the household.

Capabilities of the POLY 88 will impress you—It does far more than its compact size would suggest. The entertaining game and educational packages are but one side of it's personality. Versatile and powerful, the POLY 88 will assist you with many facets of personal finance, household routines, information storage and—what do you need? Set up your own dieting program. Analyze your energy usage. Create a space saving electronic file, accessible and useful. The possibilities are varied and far-reaching. One application may suggest many others.

The POLY 88 is a complete system from input to output. It works under home conditions, taking little space and using house current. And not only can you operate it, you can learn to program it in a short time—every aspect of the computer will be at your fingertips.

This brochure explains and demonstrates the elements of the POLY 88 system, a few of its applications, and shows an actual "run-through" of a specialized program. Please read through the brochure and take it along when you visit your POLY 88 feature dealer to operate the POLY 88.

The POLY 88 microcomputer system, for all of its abilities and potential, is a device which is perfectly suited for operation by any person—not just a computer expert. The only "language" you need to command is English!

Communication with the computer is accomplished through familiar devices. A TV monitor presents information in words and pictures, tables and graphs, economically or as fully detailed as you wish. A tape cassette recorder plays programs to the computer (or can record programs of your own invention). The standard typewriter keyboard is a convenient and universal method of directing your computer. All the elements of the system are designed for practicality and ease of use.

Actual operation of the computer is performed in a language called "BASIC", which utilizes many English words to direct its functions. Using self-explanatory statements like PRINT, RUN, PLOT, and responses like YES or NO, you will be able to converse with your system in a brief time.

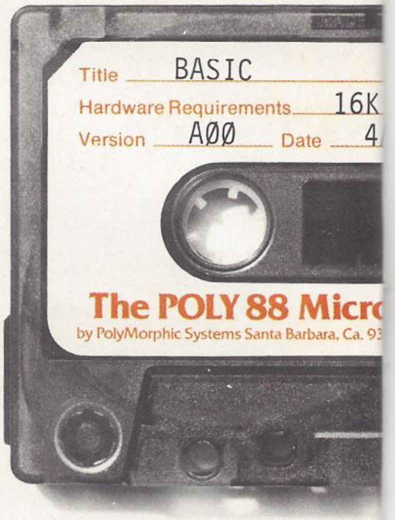
Please follow our set-up and demonstration in the following pages, and see how friendly a computer can be.

What is it like to have a computer in your home?

This "get acquainted" tour will give you the feel of using a POLY 88. The components set up compactly on a table, desk or shelf.

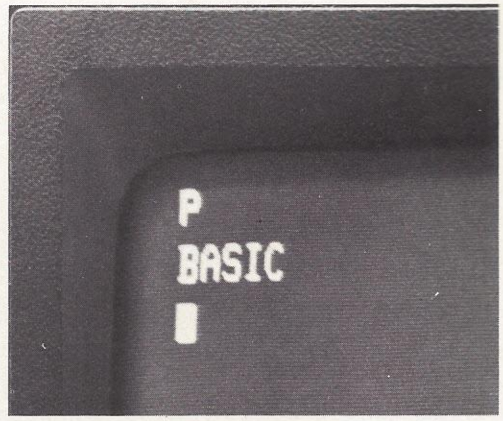
Have a seat in your favorite chair. Turn on the TV monitor and the POLY 88. The cursor, a small white rectangle, will appear on the screen.

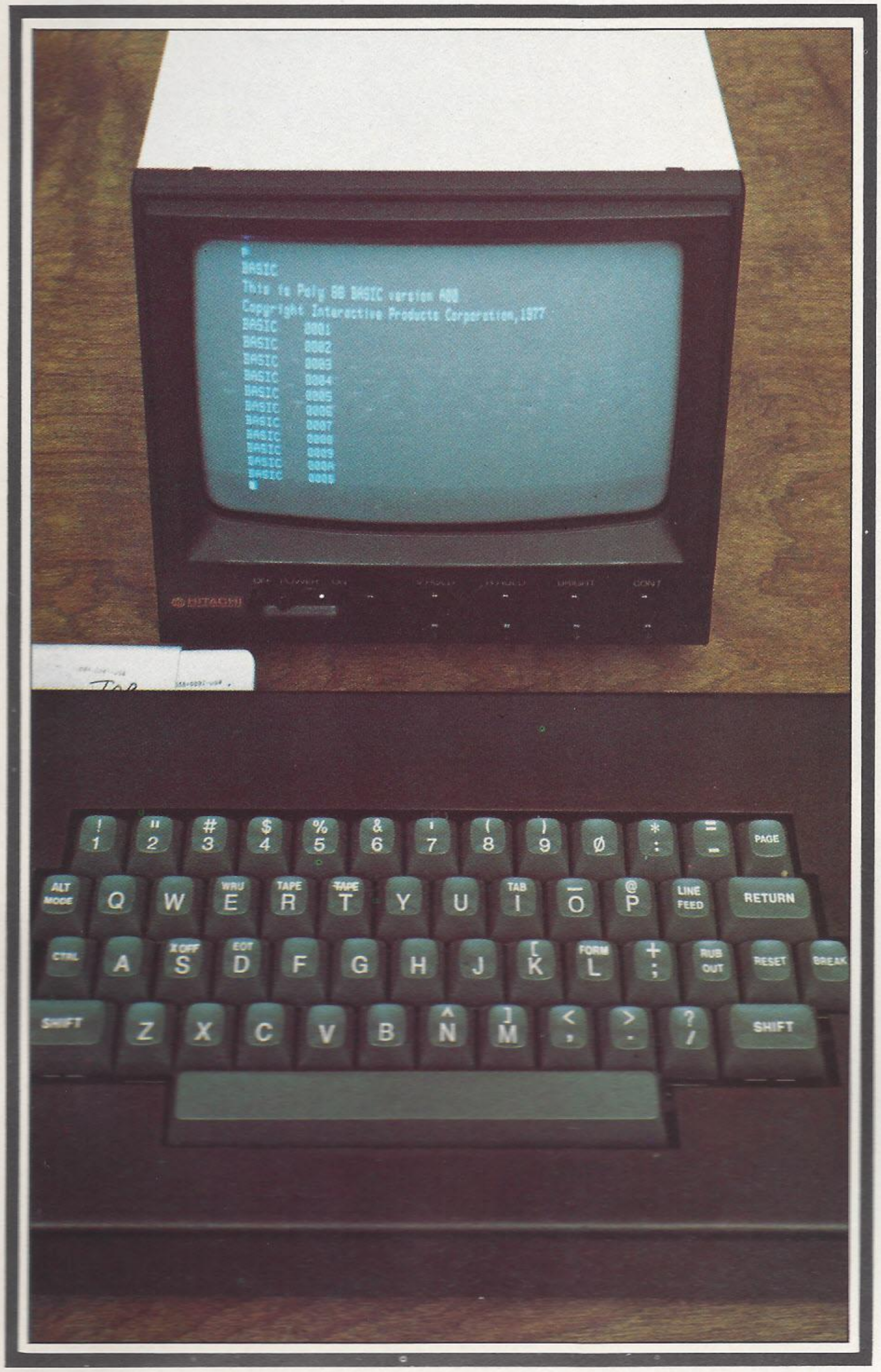
The POLY 88 is now ready to read a program from the tape cassette. Although the computer could be programmed directly from the keyboard using special "machine language" codes, these are not tailored for easy use by anyone. The language called "BASIC", on the other hand, simplifies communication with the computer by interpreting English word commands from the user. So the first step is to load "BASIC" into the computer.



LOADING BASIC INTO A POLY 88

Put the cassette labeled "BASIC" into the cassette recorder. Type the letter **P** on the keyboard. This tells the computer that it will receive a program in our "POLY" format (much faster than the "Byte" standard, which can also be read by the POLY 88). Now type the word **"BASIC".** Then depress the carriage return key. The video display reads: ⟶





The computer is "listening" for a program called "BASIC", (the interpreter language). Start the cassette in the Play mode. A message will soon appear on the screen, followed by a series of numbers (one for each data block of 256 characters) as BASIC is loaded into the computer.

When loading is complete, the screen will clear and the computer will await your instructions. When you see a message at the top of the screen followed by a "prompt" symbol (⟩), you are ready to program in BASIC.

Why use the BASIC language? Because it accepts English words and phrases as commands. The command LOAD, for example, tells the computer to read and run a variety of application programs written in BASIC. These applications range from simple information storage to comprehensive financial and statistical analysis. Simply stated, BASIC programs make full use of the enormous computing power of the POLY 88, and put *that power* in the hands of people with no previous programming experience.

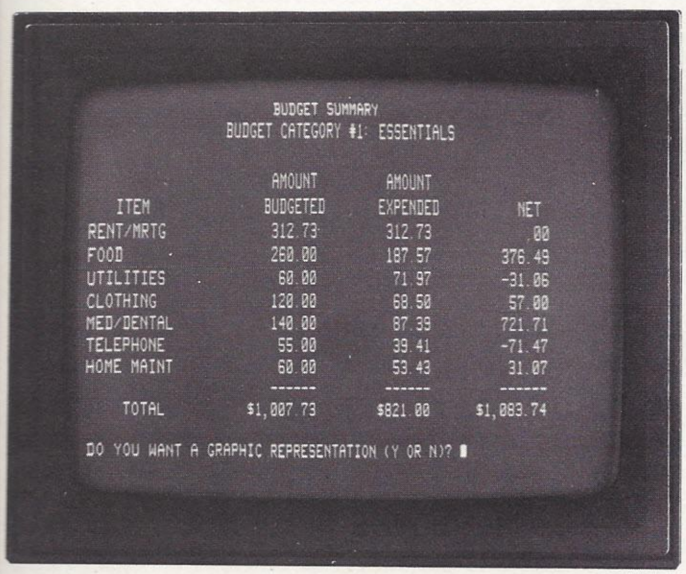
One such BASIC program is

CASHFLOW

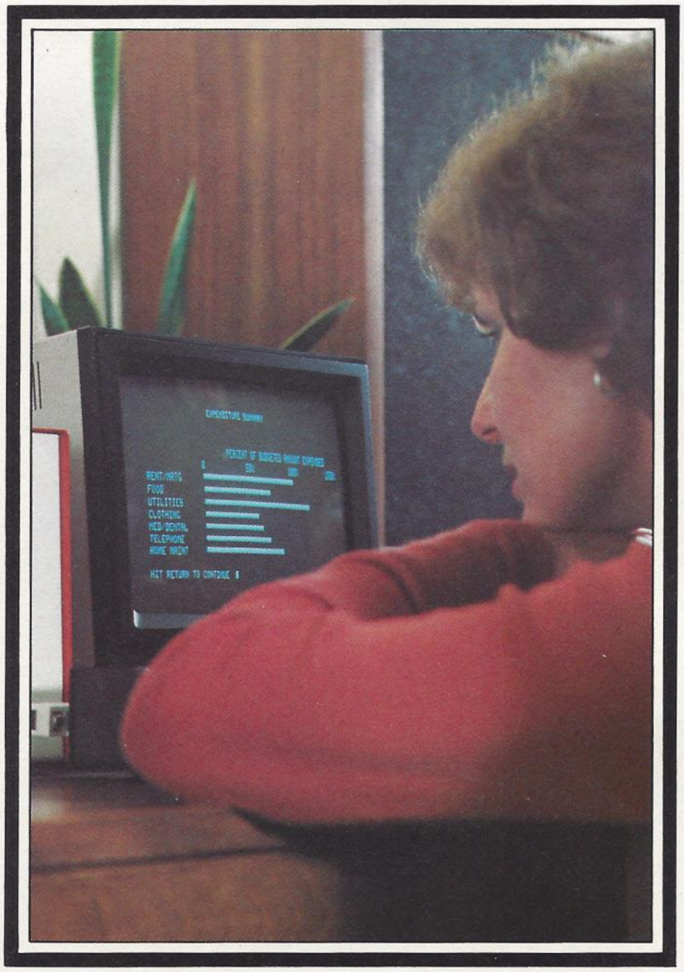
The personal financial package which is used to write budgets, record transactions and reconcile checking accounts. To load CASHFLOW, type: **LOAD, CASHFLOW, P** and hit the carriage return key.

The screen will say "Working..."and wait for the tape input. The sequence is the same as for loading BASIC. Start the CASHFLOW tape in the play mode on the recorder. As it loads, the program will print on the screen. When finished, CASHFLOW will autoexecute. CASHFLOW will ask:

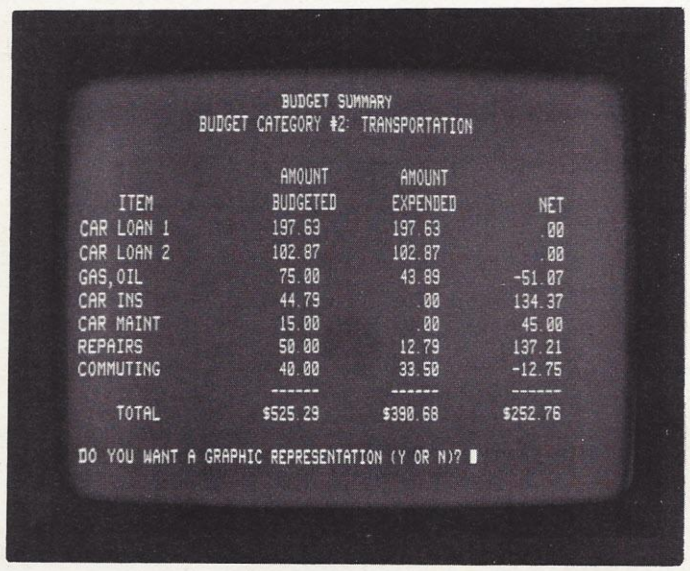
To see the financial summaries of a typical family, type **3** CASHFLOW responds:



Type **YES** (or simply "Y") to see the information in the form of a graph.



Hit a return to see the next set of budget items.



A "YES" or "NO" answer would allow you to see or bypass each remaining budget category (e.g., "FINANCIAL", "DISCRETION") or the deposit, withdrawal, income and expenditure summaries. For now, hit a return to end the summary listings, so you can see how a check or other transaction is entered.

Type **1** to see the

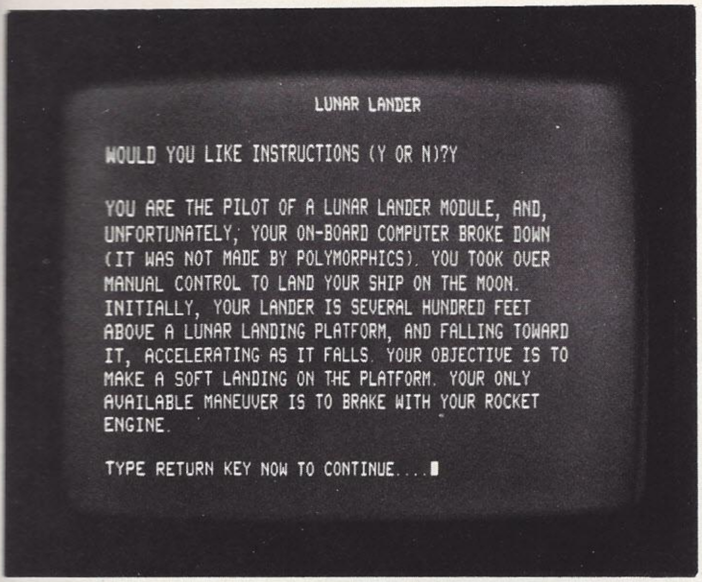
TRANSACTION ENTRY FORM

Using this form, you enter the type and amount of a new check, with automatic debiting of the corresponding budget account; or make any other transaction, including deposits, service charges, transfer of funds, etc. To review past transactions, scan by hitting the return key to go back one transaction, or by hitting the space bar to go forward one transaction.

After you have updated all your transactions, you would record the program on a new tape for use the following month, and save each old tape to form a complete financial record.

You don't have to use CASHFLOW to handle your finances—write your own program, tailored and customized to *your* circumstances.

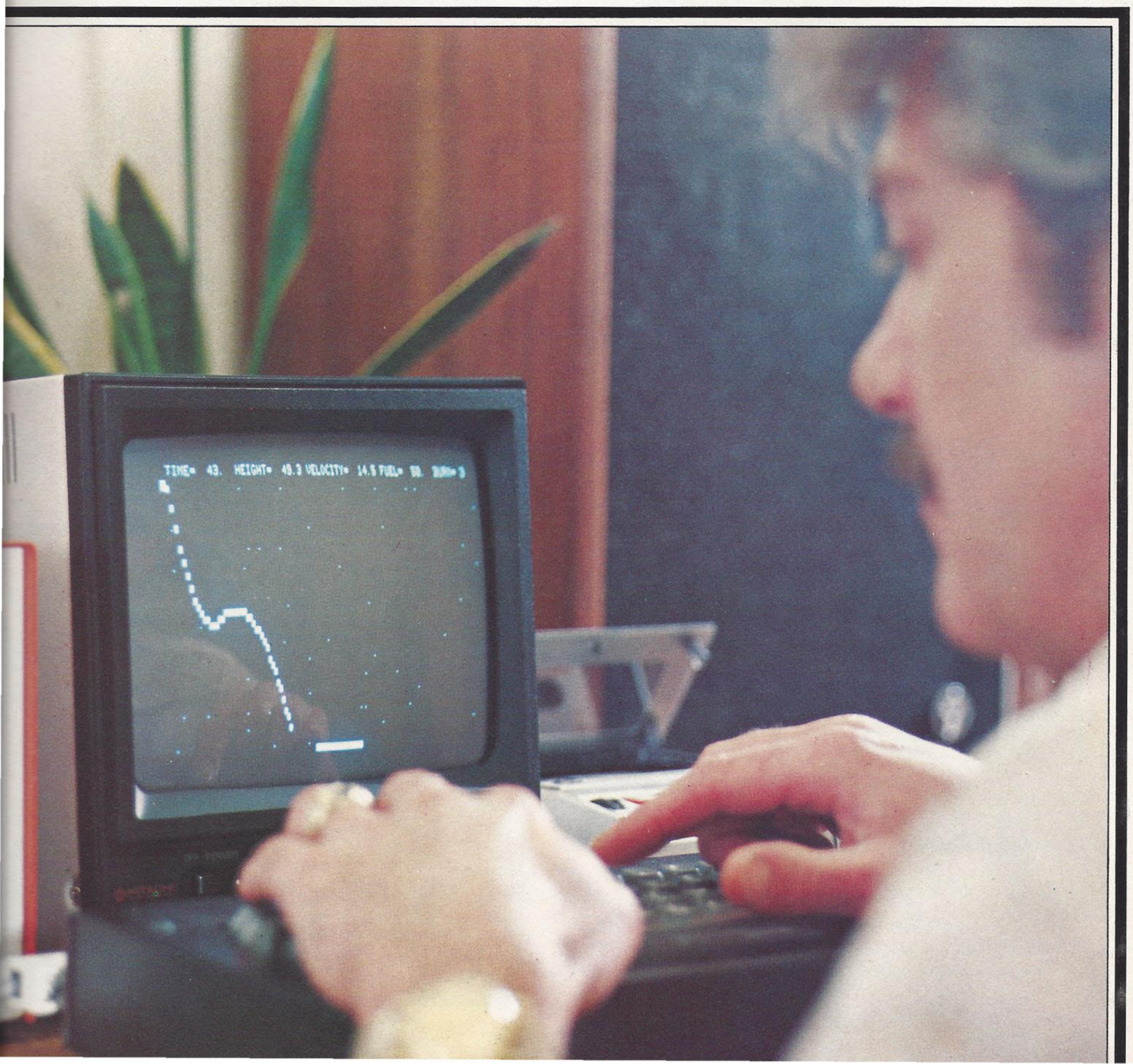
But POLY 88 is not just dollars and cents. The whole family will enjoy fascinating animated computer games. For example, LANDER...



GAM

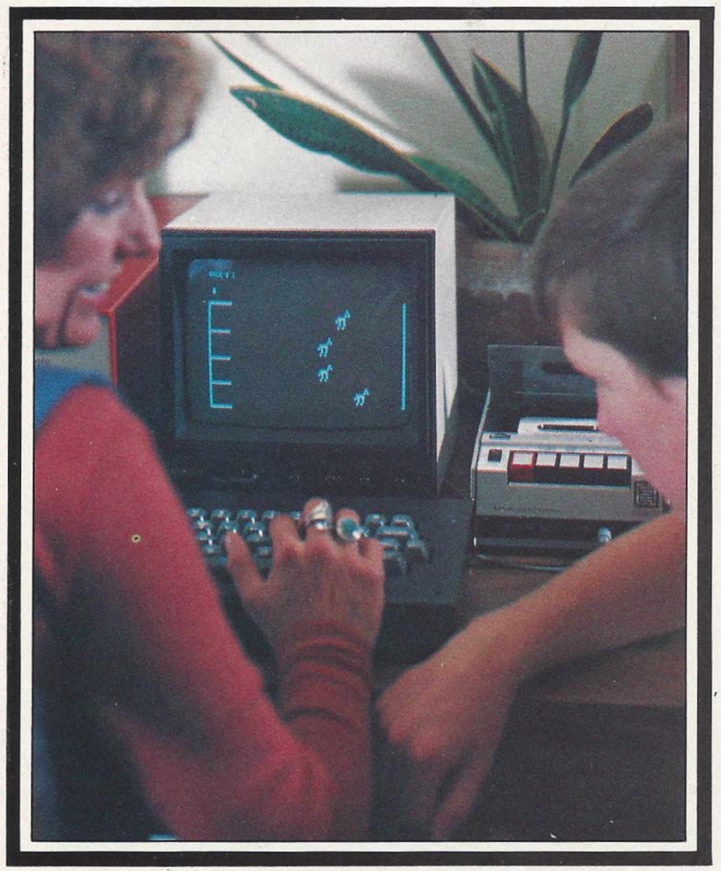
LANDER is a game program which tests your skill as a lunar astronaut. You must use your retro-rocket skillfully and efficiently to land safely on the moon.

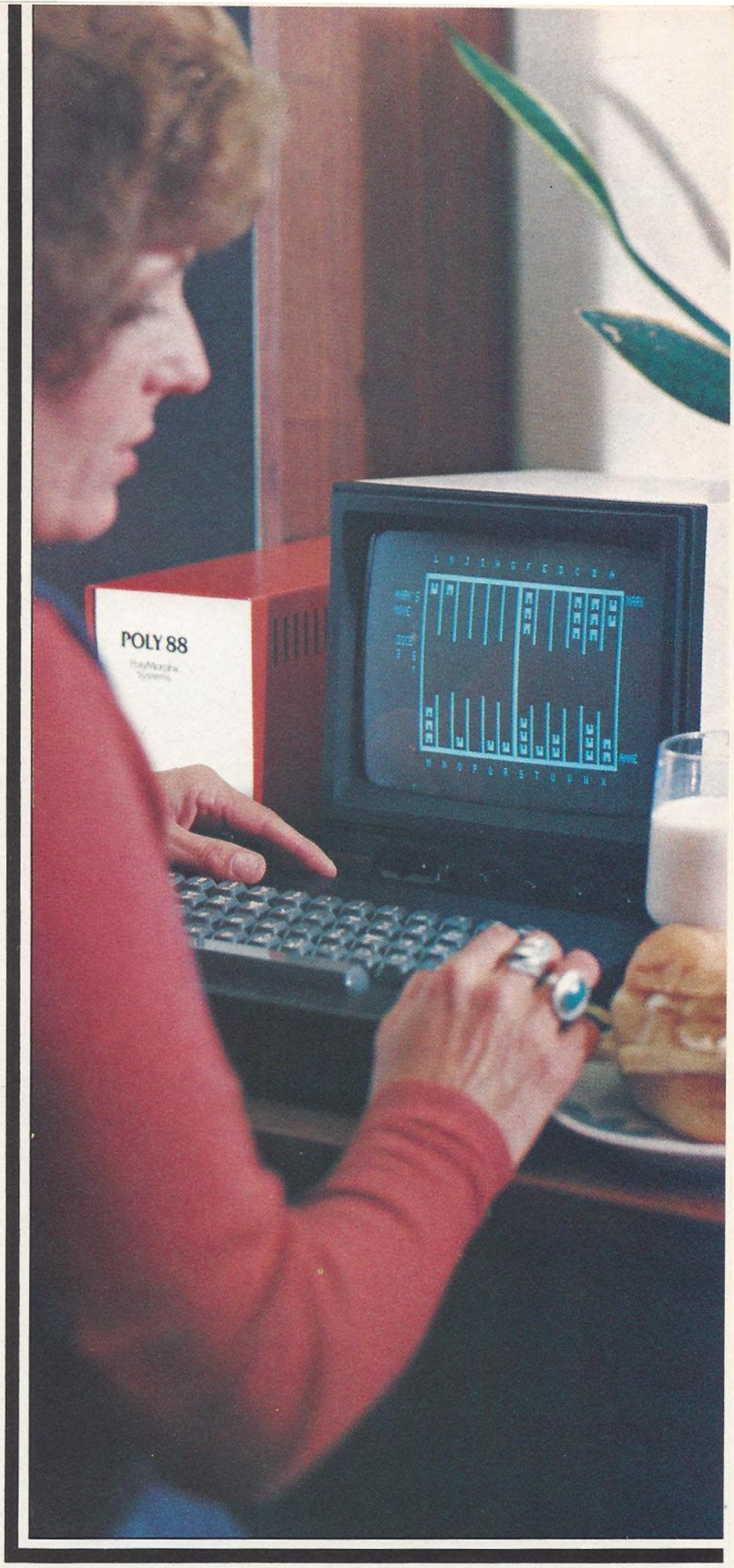
To load LANDER, follow the simple steps. Type **LOAD, LANDER, P** hit the return key. The computer answers "Working..."; Play the LANDER tape. After the data blocks finish loading on the screen, the program will start. It will ask if you want instructions. Type **YES** and hit return. The game instructions will appear on the screen. To land safely, you must select and vary the rocket thrust, keeping an eye on altitude, speed and fuel consumption. Each game is slightly different. Miss the platform and you crash—and get a free critique of your flying skills from the computer! This game will give you hours of fun and excitement...

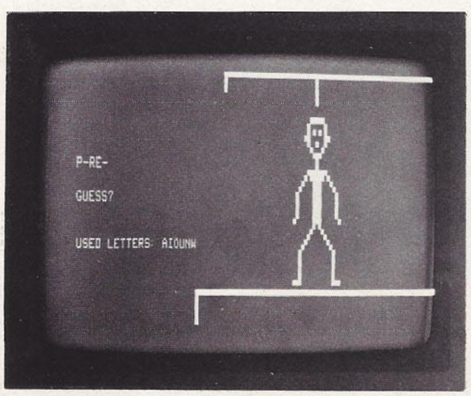


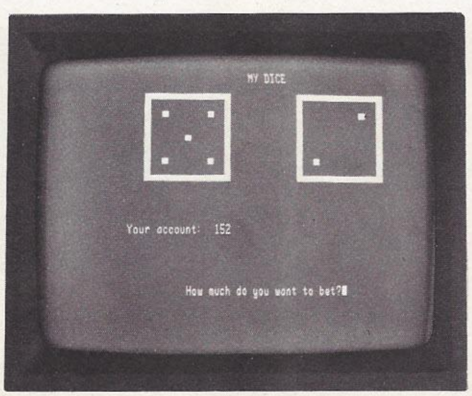
MES

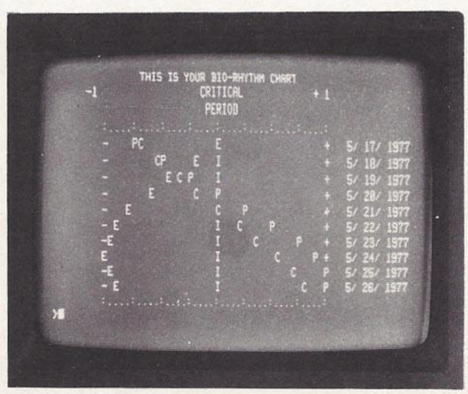
...or sample our other challenging games, like BACKGAMMON...test your spelling ability under pressure with the HANGMAN...break the bank shooting CRAPS (honest dice, no house odds)... cheer home a derby winner with HORSE...get in touch with your biorhythms with BIOCYCLE...or any of our other fascinating games. They are easily used by the novice, and help develop your programming skills. For added enjoyment, devise your own games! There is nothing quite like the satisfaction of programming your own diversions, with personalized messages, varying difficulties, and differing strategies. The POLY 88 can be a very educational companion, growing with a child or jogging an adult's mental skills. It's easy with our BASIC language—so simple and straightforward that you will feel right at home with it in a few weeks.











LEARNING THE BASICS

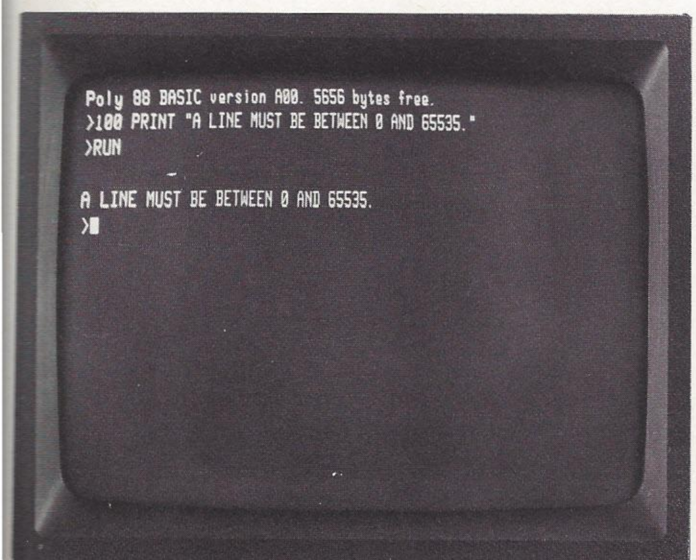
Now that you have seen the range of the POLY 88's abilities using our applications, you're ready to see what it can do in your hands. *What, you've never run a computer until right now?* That's all right; no experience is necessary to become proficient in BASIC. The commands are simple and direct. For example, type **SCRATCH** and hit return. You have just erased the last game or application in the machine (but you are still in the BASIC language), and you are ready to program.

A program is a series of statements and directions written in lines. To let the computer know you are programming, give it a line number (the number lets the computer index the information and makes it easier for you to refer to or repeat that line later).

For instance, type:

100 PRINT "A LINE NUMBER MUST BE BETWEEN 0 AND 65535."

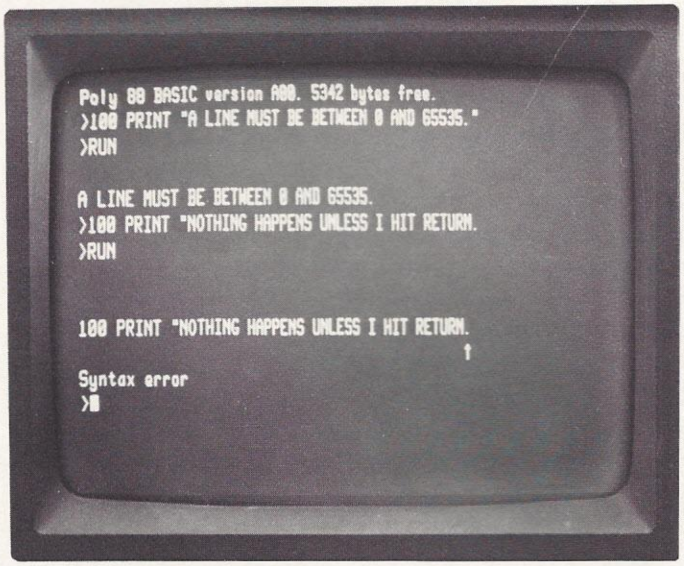
and hit return. Type **RUN** and hit return again. The video display shows:



You have just programmed the computer—to print a line that you specified. Try this one. Type:

100 PRINT " NOTHING HAPPENS UNLESS I HIT RETURN.

Once again, hit return, type **RUN** and hit return again.



The computer does not understand this instruction because it was typed incorrectly. Compare it to your first program, and retype it as you think it should be. The computer will "prevent" you from making errors in this fashion by calling attention to the error (in this case, an error of syntax or language use). Suppose you catch your own error as you are typing the line? Try it. Type:

100 PRINT "USE THE DELTE KEY

To correct, hit the key labeled DELETE until the cursor backs up over the error. Then type the correct version. Your finished line should read:

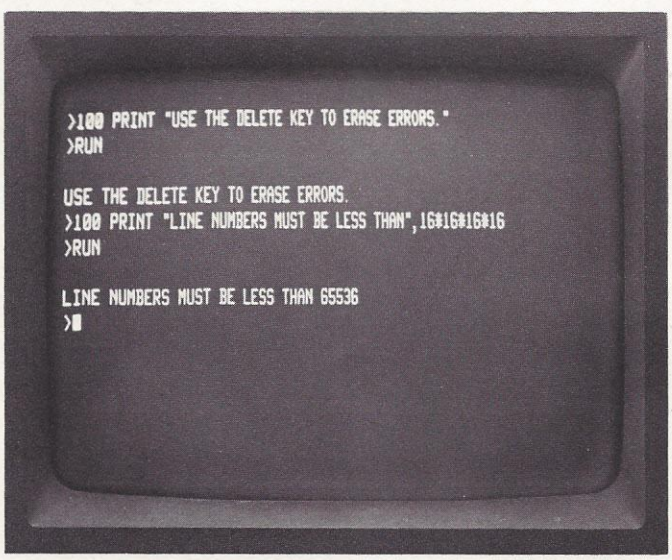
100 PRINT "USE THE DELETE KEY TO ERASE ERRORS."

Hit return, type **RUN,** hit return again.

Congratulations! You are now experienced, with three programs to your credit. Now try something a bit different. Type:

100 PRINT "LINE NUMBERS MUST BE LESS THAN", 16*16*16*16

Run the program (hit return, type RUN, hit return).



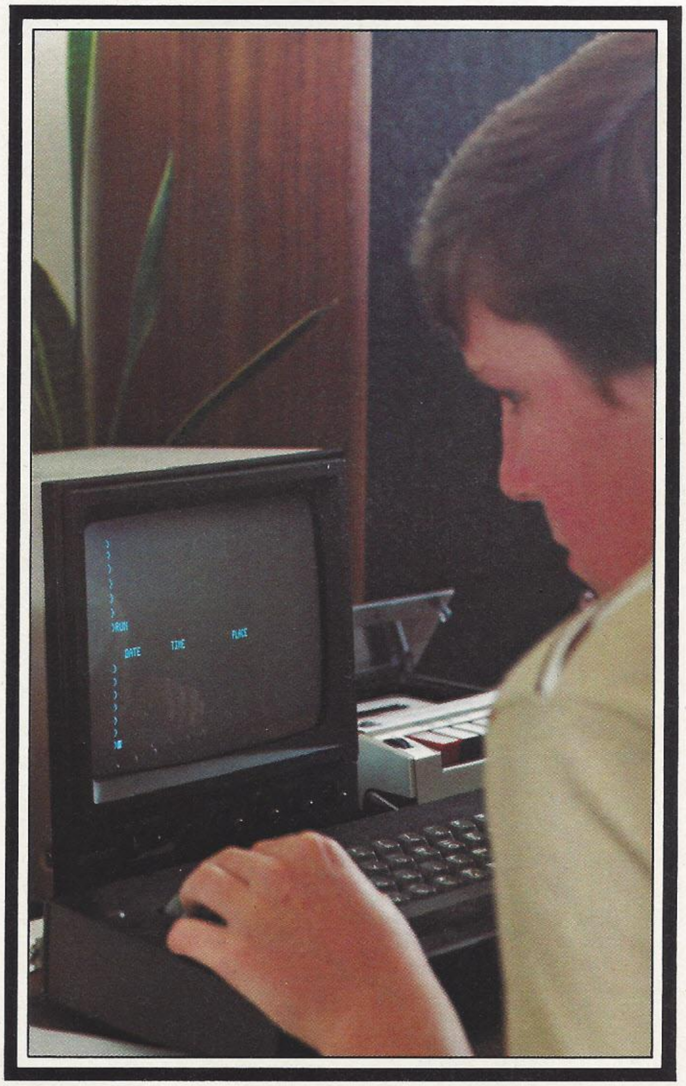
As you can see, it's done a multiplication problem. Now type **100 N=16*16*16*16** (hit return) **200 PRINT "LINE NUMBERS . MUST BE LESS THAN", N**

Run the program (return, RUN, return). This program adds a variable—that is, "N" represents the number, whatever you assign to it. The number asigned to "N" was set in line 100, and used in line 200. Any letter can be used to represent a number this way.

Now, put column headings in their places by typing:

100 PRINT TAB (3), "DATE", TAB (15), "TIME", TAB (34), "PLACE"

Run the program.



Typing the word TAB tells the computer how much to space over from the left margin before printing the word.

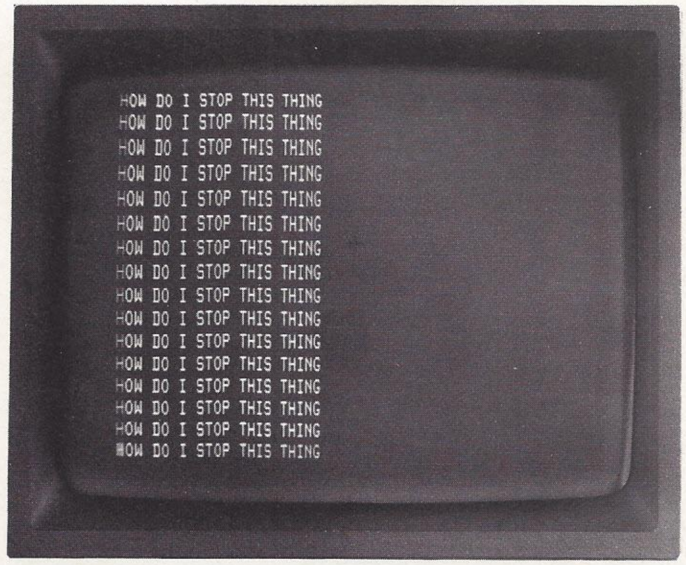
Want to repeat a line more than one time? Just refer back to it in a later line. This two-line program does just that.

100 PRINT "HOW DO I STOP THIS THING?"

Hit return. Type:

200 GOTO 100

Run the program. Following your instructions, it will repeat itself. . . Hit a few keys. Now before you get desperate to stop it, hold down the key labeled CNTRL and hit Y. This will always stop a program, and return you to BASIC.



Try a few more programs. Get fancy . . . have the computer call you by name. When your name comes up, have it give you a title. May we suggest the title of "Programmer?

To delete this last program,

Type SCRATCH

The last program you did simply returned the computer to a preceding line, printing it over and over. Suppose you want the computer to print something a specific number of times, then stop?

THE PROBLEM: Have the POLY 88 print a statement five times. THE PROGRAM:

100 PRINT "THE POLY 88 CAN COMMUNICATE IN BASIC."

(this line tells the computer what to print)

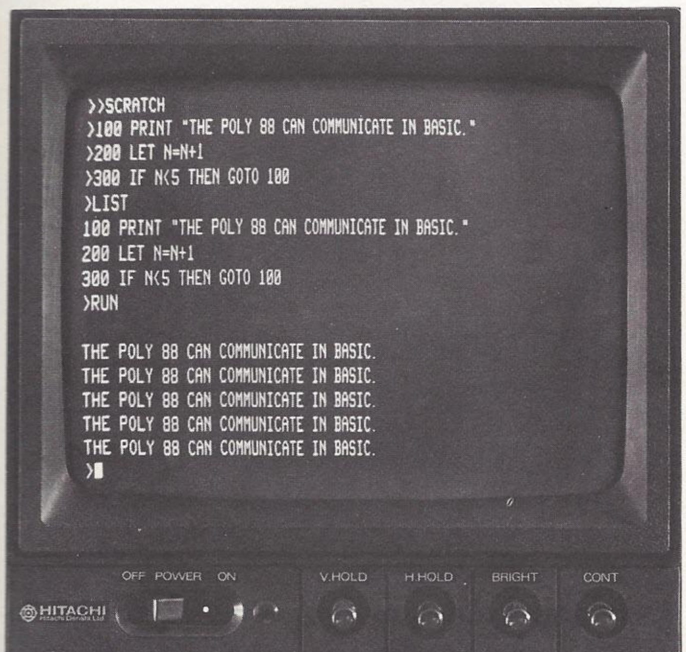
200 LET N=N+1

(N is the counter. Each succeeding time the computer goes through the program, the value of N will increase by 1)

300 IF N<5 THEN GOTO 100

(If the value of N is less than 5, return to line 100 and print the statement again.) NOTE: when N reaches 5, the computer will not return to line 100, but will go to a new line. Since there are no further lines, the program ends.

Once the program is typed in, hit return, type **LIST** and hit return again . . . to show the program, as you typed it, in its entirety. Now RUN the program.



Using the BASIC commands covered on the last few pages some very powerful programs can be developed. "The Manhattan Indian Problem", which follows, will demonstrate this, and will tell you once and for all whether New York was a good buy!

THE MANHATTAN INDIAN PROBLEM

In 1627 Peter Minuit bought Manhattan Island from Chief Powhatan for approximately $24.

THE PROBLEM: If the $24 had been deposited in a savings account, what would be the value of the account? Calculate the answer for 50 year intervals from 1650 to 2000, using interest rates of 1%, 2%, 3%, and 4% compounded annually.
THE PROGRAM: The following is one way to solve the problem. The program has two parts, the first of which sets up a tabular display of the information. To ready the POLY 88, type SCRATCH and hit return.

Type the five lines below into the computer, then RUN the program. The screen should look like this: ⟶

If an error is shown, compare the line with the corresponding line below. To correct it, retype and enter only that line.

100 PRINT TAB (20), "BALANCE ON $24 DEPOSITED IN 1627 ($)"

(this centers the title for the table)

150 PRINT TAB (26), "ANNUAL INTEREST RATE"

(this centers the title for the columns)

200 LET I=I+1

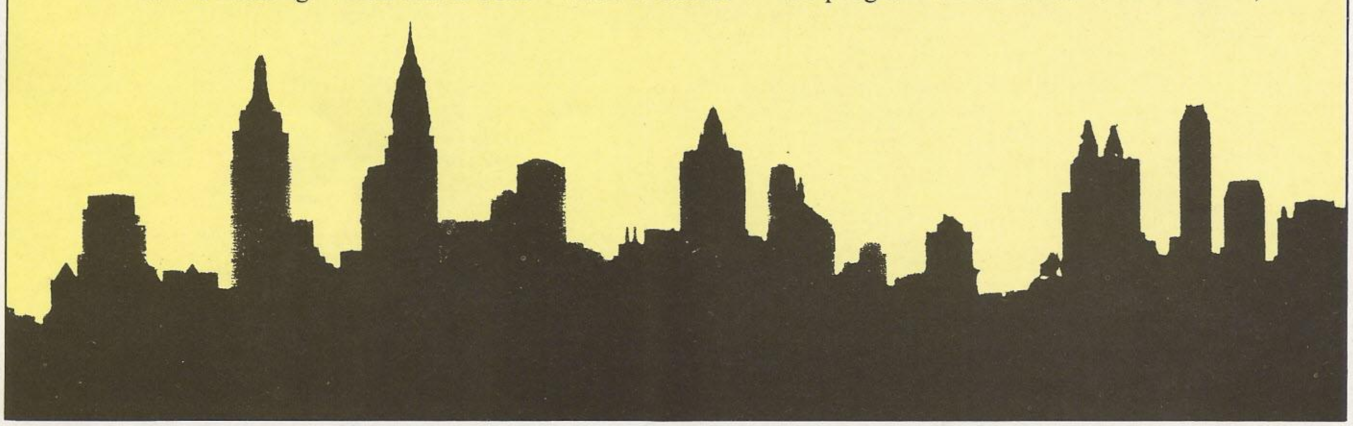
(this causes I to increase its value by 1 with each succeeding run-through of the program. First time through, I=0)

250 PRINT TAB (13*I+4), I, "%",

(this puts the proper number over each column one at a time. The comma after % will cause the next line bearing a PRINT instruction to print on the same line again and again as directed)

300 IF I<4 THEN GOTO 200

(this sends the computer back to line 200 each time it gets to line 300, until all columns—4 of them—have a heading. When this is done—when I reaches 4—the program will continue to the next line)



The second part of the program is

THE CALCULATING SECTION

350 PRINT

(the cursor has been staying on the same line because of the comma at the end of line 300. This lone PRINT statement moves the cursor on to the next line.)

400 LET Y=1600

(the letter Y will represent, and count, the years. Remember, variables will start at zero unless otherwise assigned. Here we are assigning 1600 as the beginning year)

450 LET Y=Y+50

(each succeeding value of Y will be greater by 50, to provide our 50-year intervals)

500 LET I=0

(we used several values of I previously, up to I=4, to lay out the table in line 250. Now it is started again for calculating and formatting. If I=1, the program prints the first column; I=2, the second; and so forth)

550 PRINT Y,

(this prints the year in the left column. As in line 250, the comma after Y keeps the cursor on the same line, to print the appropriate value in each of the 4 columns)

600 LET I=I+1

(this line updates I by 1 (percent) on each succeeding run-through, to do each column in order)

650 PRINT TAB (13*I), 24* (1+I/100)^(Y-1627),

(this line does the calculation using the formula for compounding interest. It then prints the answers in the proper column) The symbol ^ is used for exponentiation (e.g., X ^ Y means X raised to the Y power)

700 IF I<4 THEN GOTO 600

(this creates a "loop" between lines 600 and 700, to calculate and print the 4 values of I for each year—one per column)

750 PRINT

(this brings the cursor down to the next line (see 350)

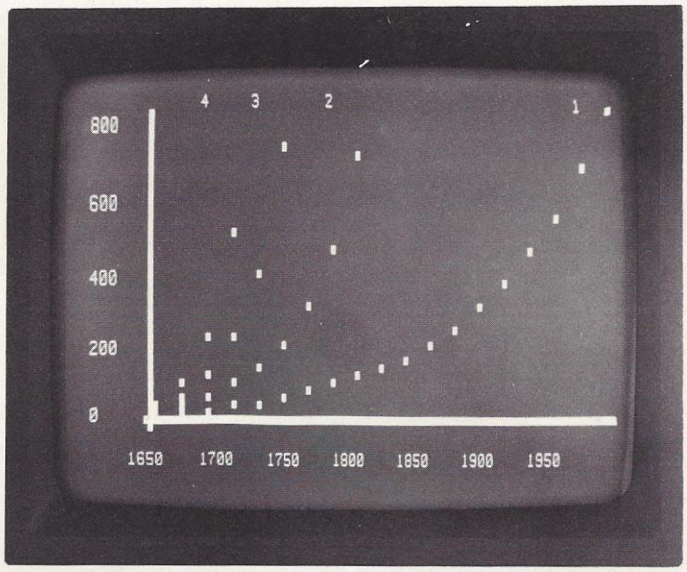
800 IF Y<2000 THEN GOTO 450

(the calculating loop between lines 600 and 700 does several interest problems for each year. This second loop between 450 and 800 causes the computer to continue calculations up to the year 2000. After Y reaches 2000—there being no further lines—the program stops.

Now RUN the program, and you will see:

The following program demonstrates further capabilities of BASIC—graphic representation. It uses some terms we have not covered — PLOT, FOR, NEXT, EXIT, for example—so we won't explain it in detail. It is presented to show that the full power of the POLY 88 can be exploited using the BASIC language.

No other microcomputer has these graphics capabilities.




```
10 !CHR$(12)\FOR D=800 TO 0 STEP -200\!\!D\!\NEXT
20 FOR T=1 TO 7\!TAB(8*T-4), 1600+50*T,\NEXT\!CHR$(11)
30 FOR Y=6 TO 45\PLOT 15, Y, 1\NEXT
40 FOR X=14 TO 126\PLOT X, 7, 1\NEXT
50 IF I<4 THEN I=I+1 ELSE POKE 63552, 127
60 FOR X=0 TO 111 STEP 6\Y=1.08*(1+I/100)^(23+3.125*X)
70 IF Y<41 THEN PLOT X+16, Y+7, 1\NEXT
80 PLOT X, 47, 0\!I\EXIT 50
```

If you've looked all over the keyboard for \ , try "FORM" (shift L).

In only 8 lines, the screen is cleared, plotting axes with appropriate scales are drawn, and the results of the Manhattan table are plotted as a graph.

Type this program in line for line, and see how the resulting graph looks. It is not necessary to understand how the program works to realize the great potential for presenting information in different ways.

You have demonstrated to yourself the power of BASIC to solve difficult, specialized problems with relative ease. Look how far you've come—and you've barely scratched the surface! Your POLY 88 dealer will be happy to show you other features of this outstanding computer, its simplicity and its reasonable price. You're probably dreaming up your own applications already—now make them a reality.

SPECIFICATIONS

Chassis

Power Supplies: +9V at 6A, +18 at 0.75, –18 at –.25A (unregulated) per chassis.

Number of cards: 5

Compatibility: Accepts POLY 88—/IMSAI/Altair cards.

Bus Connector Type: Power switch, reset button, halt indicator and power indicator.

Cabinet Dimensions: 4¼" w X 6¾" h X 17" d.

Expandability: Up to 4 chassis may be plugged together.

Processor Card

Processor Type: 8080A

Clock: Crystal controlled 1.8432MHz.

Number of Instructions: 78 with 244 variations.

CPU Registers: 10

Addressing Modes: Direct, immediate, register, register indirect.

Addressing Range: 65, 536 bytes of memory and 256 ports.

Interrupts: 8 level vectored.

Real Time Clock: 50/60 Hz (referenced to line frequency).

Bus Loading: Outputs drive 30 TTL loads, inputs are 1 TTL load or less

Power Requirements: +8 to 10V at 1.2 amps max., +16 to 20V at 200 mA. Max., –16 to 20V at 140 mA. max. (inc. 3 2708 EPROM).

RAM: 512 bytes 500 nesc. access time.

ROM: Sockets provided for 3K bytes of EPROM or (2708 type) ROM.

Serial I/O Port (option)

Baud Rate: Software controlled 12.5 to 9600 baud (asynchronous) 800 to 57,600 baud (synchronous).

Data Format: 5, 6, 7, or 8 bits with or without parity.

Ports: Can handle up to 2 minicards.

Power Requirements: +8 to 10V at 150 ma, –16 to 20V at 35 ma.

Video Terminal Interface

Character Font: 7 X 9 matrix alphanumeric and 6 element graphic characters.

Character Set: 96 character ASCII plus 32 special characters.

Number of Characters: 1024; 64 characters per line with 16 lines.

Graphic Resolution: 128 horizontal by 48 vertical.

Video Output: Composite video negative sync.

Required Video Bandwidth: 5.5 MHz.

Input Port: 8-bits plus positive or negative strobe.

Power Requirements: +8 to 10V at 1.25A typ., +16 to 20V at 30 ma max. –16 to 20V at 20 ma max., not including keyboard requirements.

Byte PolyPhase Cassette Interface

Baud Rate: 300, 600 (Byte); 1200, 2400 (Polyphase).

Connector (to recorder): Standard 25 pin "D" type female Mating connector type=DB-25p or equivalent.

Connector type (to CPU): 14 wire ribbon cable with DIP plug termination.

Power Requirements: +5 ±0.25 VDC at 150 ma max., –5 ±0.25 VDC at 20 ma max.

Input: Approximately 2Vp-p into 15 Ω

Motor Controls: (2) will switch up to 200 mA at up to 30V (negative ground).

8K RAM

Storage Capacity: 8192 X 8 bits

Memory Type: 91L02 or equivalent.

Access Time: 500 ns. max.

Address Decoding: Switch selectable in 8K increments.

Memory disable: Controlled by bus pin 67 (PHANTOM).

Power Requirement: +8 to +10 at 2.6A max. +2.2V 1.5A max. (standby).

Poly I/O Ideaboard

I/O Addressing: Blocks of four addresses.

Power Requirement (excluding user added components): +8 to 10V at 370 mA. max.

Firmware Monitor

Size: 1K bytes in 2708 type read-only memory ROM or compatible mask programmed ROM.

Operation: Fits in zero-addressed CPU ROM socket. Runs on system power-up or front panel reset.

Functions: Tape loader for Byte standard 300 baud or Polyphase 2400 baud encoded Polyformat absolute binary cassette files. Front panel simulator program produces register and memory display on system video display. Commands allow register, memory modification, program interrupt and single step or return to interrupted program.

Utilities: Routines for teletype simulated I/O on the memory mapped video display, character fetching from system keyboard, hexadecimal conversions, real time clock.

Features: "Wormholes" allow reassignment of I/O devices independent of program I/O handling. Video driver recognizes TAB, FORMFEED, BACKSPACE, LINE ERASE, VERTICAL TAB and CARRIAGE RETURN. Address of video display is reassignable. Interrupts handled through a vector table which allows reassignment of service routine addresses. Real time clock has 2 year delay capability with automatic execution of preset task on timeout.

11K BASIC by PolyMorphic Systems

Size: 11K bytes.

Scientific Functions: Sine, cosine, log, exponential, square root, random number, x to the y power.

Formatted Output

Multi-line Function Definition

String Manipulation and String Functions

Real-Time Clock

Point-Plotting on Video Display

Array Dimensions limited by Memory

Cassette Save, Load and Verify of Named Programs

Multiple Statements per Line

Renumber

Memory Load and Store

8080 Input and Output

If Then Else

Input Type-Ahead

Commands: RUN, LIST, SCR, CLEAR, REN, CONT, SAVE, LOAD, VERIFY.

Statements: LET, IF, THEN, ELSE, FOR, NEXT, GOTO, ON, EXIT, STOP, END, REM, READ, DATA, RESTORE, INPUT, GOSUB, RETURN, PRINT, POKE, OUT.

Built in Functions: FREE, ABS, SGN, INT, LEN, CHR$, VAL, STR$, ASC, SIN, COS, RND, LOG, TIME, WAIT, EXP, SORT, CALL, PEEK, INP, PLOT.

Printer Interface

Interface Standard: Full RS-232 or current loop (hardware selectable).

Required Input From CPU: TTL level serial data, clock, port select, and control

BAUD Rate: 57,600 max.

Output to CPU: TT level serial data, clock and control signals.

Connector Type (to serial device): EIA standard 25 pin female. Mating connector type + DB-25P or equivalent.

Connector Type (to CPU): 14 wire ribbon cable with DIP plug termination.

Dimensions: 2.1" X 4.0" (5.33 cm X 10.16 cm).

Storage Temperature: –25 to +85 C.

Operating Temperature: +5 to +55 C.

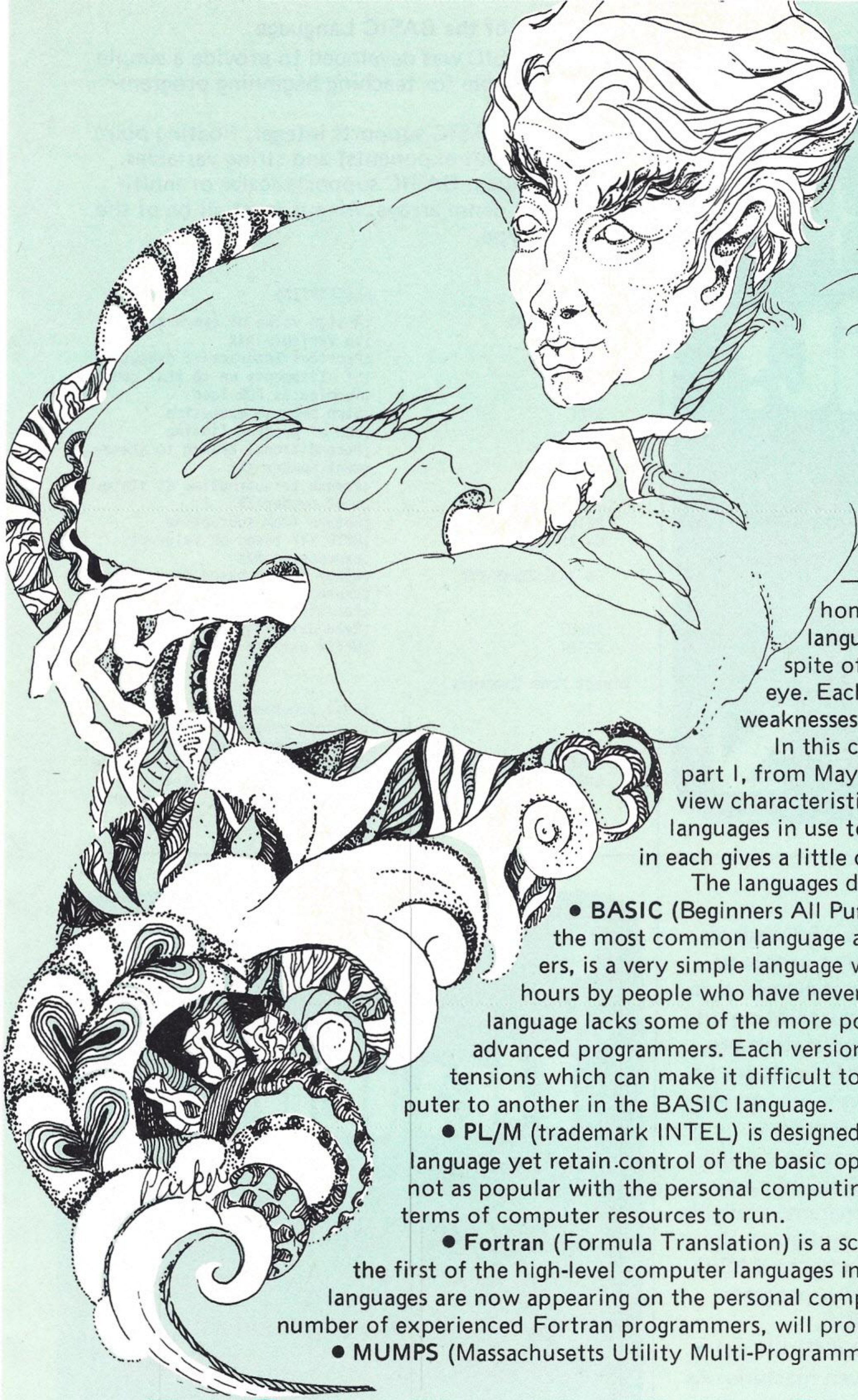
Humidity: 0 to 95% non-condensing.

Power Requirements: +5 +0.25 VDC at 42 ma max. +12 +0.6 VDC at 24 ma max., –12 +0.6 VDC at 23 ma max.

PolyMorphic Systems

460 Ward Drive Santa Barbara, Ca. 93111 (805) 967-0468

YOUR PERSONAL GENIE
PART 2



by Tom Munnecke

The patterns of language we use in communicating with a computer are similar in many ways to the spoken language we use in our everyday lives. It is the medium used by the programmer to give the computer its personality — be it a game, a tax assistant or a home burglar alarm. These computer languages are not difficult to learn in spite of cryptic appearance to the untrained eye. Each language has its own strengths and weaknesses for various applications.

In this continuation of the Personal Genie, part I, from May/June Personal Computing, we review characteristics of some of the more common languages in use today. A sample program coded in in each gives a little of the flavor of the language.

The languages described are:

- **BASIC** (Beginners All Purpose Symbolic Instruction Code), the most common language available on today's home computers, is a very simple language which may be learned in a few hours by people who have never programmed before. However, the language lacks some of the more powerful features desired by more advanced programmers. Each version of the language has its own extensions which can make it difficult to transfer a program from one computer to another in the BASIC language.
- **PL/M** (trademark INTEL) is designed to give the programmer a high-level language yet retain control of the basic operations of the computer. PL/M is not as popular with the personal computing set because it is very expensive in terms of computer resources to run.
- **Fortran** (Formula Translation) is a scientific language which was one of the first of the high-level computer languages in the late 1950's. Subsets of the languages are now appearing on the personal computer market and, because of the number of experienced Fortran programmers, will probably become very popular.
- **MUMPS** (Massachusetts Utility Multi-Programming System) is a language recently

A Summary of the BASIC Language

Purpose: BASIC was developed to provide a simple language suitable for teaching beginning programming students.

Data Types: BASIC supports integer, floating point (with or without exponents) and string variables.

Data Structures: BASIC supports scalar or multidimensional dense arrays. Arrays must all be of the same data type.

Commands:

NAME	DESCRIPTION
LET XXX=YYY	;Assign value of expression YYY ;to variable XXX
FOR	;Provides interactive execution ;of statements up to NEXT command
NEXT	;Terminates FOR loop
STOP	;Stop program execution
END	;End of program listing
GOTO XXX	;Unconditional branch to statement number XXX
GO SUB XXX	;Branch to subroutine at statement number XXX
RETURN	;Return from subroutine
ON XXX GOTO YYY	;GOTO YYY based on value of ;expression XXX
ON XXX GOSUB YYY	;GOSUB to YYY based on value of ;expression XXX
IF	;Permits conditional execution
INPUT	;Read data from console device
PRINT	;Write data to console device

Direct Mode Commands

LIST	;List program lines
RUN	;Execute program
NEW	;Erase program from memory
SAVE	;Save program on storage device
LOAD	;Load program from storage ;device. Clears out old program ;test

standardized by the National Bureau of Standards to handle interactive data management problems. MUMPS is unique because it combines elements of what other computer systems call an operating system, file management system and application programming language in one language. Subsets are beginning to appear on the market, especially on the mini-computer compatible micro computers.

- **FOCAL** (From DEC) is an interactive calculation-oriented language similar to BASIC in many ways. There are a fair number of programs available in the language, mostly taken from minicomputer systems. Because of FOCAL's similarity to BASIC and BASIC's popularity, most FOCAL programming will probably give way to BASIC.
- **Assembler** language is the most difficult yet powerful language available on a given machine. As discussed in Part I, assembler language has good points in efficiency and a drawback in programming difficulty.

These languages are not by any means a complete listing of the spectrum of computer languages. Space limitations force exclusion here of all but the most common languages of interest to the personal computer programmer. The author is interested in reports of other languages in use, especially new languages. He can be reached through the Personal Computing Editorial offices.

```
APPEND                 ;Load program from storage
                       ;device without clearing out
                       ;old one
TRACE ON               ;Print each line number as it
                       ;is executed
TRACE OFF              ;Turns trace off
```

Functions

```
RND (XXX)              ;Random number
TAB (XXX)              ;Move to print position XXX
INT (XXX)              ;Greatest integer less than XXX
ABS (XXX)              ;Absolute value of XXX
SGN (XXX)              ;Sign of expression
POS (XXX)              ;Position of print load
LEN (XXX)              ;Length of string
ASC (XXX)              ;ASCII numeric equivalent of XXX
CHR$ (XXX)             ;Inverse of ASC
VAL (XXX)              ;Convert from string to numeric
STR$ (XXX)             ;Convert from numeric to string
LEFT$                  ;Extract leftmost characters
                       ;of a string
RIGHT$                 ;Extract rightmost characters of
                       ;a string
MID$                   ;Extract characters from middle
                       ;of a string
SIN, COS, TAN          ;Sine, Cosine, and Tangent
EXP                    ;Exponentiation
SQR                    ;Square root
DEF                    ;Define user functions
USER (XXX)             ;Call assembly language routine
                       ;at location XXX
```

This is a sample BASIC program to take an average of student grades, then print out the averages. BASIC is the most popular language used on personal computers today. Source: ***Programming Languages, PDP-8 Family Computers,*** **Digital Equipment Corp.**

```
10    REMARK - PROGRAM TO TAKE AVERAGE OR
15    REMARK - STUDENT GRADES AND CLASS GRADES
20    PRINT "HOW MANY STUDENTS, HOW MANY GRADES PER
      STUDENT";
30    INPUT A,B
40    LET I=0
50    FOR J=I TO A-1
55    LET V=0
60    PRINT "STUDENT NUMBER =";J
75    PRINT "ENTER GRADES"
80    FOR K=D TO D+B-1
81    INPUT G
82    LET V=V+G
85    NEXT K
90    LET V=V/B
95    PRINT "AVERAGE GRADE =";V
96    PRINT
99    LET Q=Q+V
100   NEXT J
101   PRINT
102   PRINT
103   PRINT "CLASS AVERAGE=";Q/A
104   STOP
140   END
RUN
```

A Summary of the FORTRAN Language

Purpose: FORTRAN is a language designed for scientific computation problems. It is usually compiled.

Data Types: FORTRAN supports Integer and Floating Point numbers in addition to logical (true/false) and Hollerith (characters).

Data Structure: FORTRAN supports scalar and single dimensional dense arrays. Items may share the same memory space with the common and equivalence statements.

Statements:

```
NAME                   DESCRIPTION

COMMENT                ;for documentation
CONTINUE               ;control passes to next
                       ;statement
GOTO XXX               ;transfer control to state-
                       ;ment number XXX
```

```
IF                        ;permits conditional execution
DO XXX                    ;allows looping according to
                          ;arguments of XXX
PAUSE XXX                 ;display XXX, then temporarily
                          ;halt program
STOP                      ;stop program execution
END                       ;end of program listing
READ XXX                  ;read data from device
                          ;specified in XXX
WRITE XXX                 ;write data to device
                          ;specified in XXX
FORMAT                    ;specify format for READ
                          ;or WRITE statements
COMMON                    ;define common area for data
                          ;storage
FUNCTION                  ;define a user-defined
                          ;function reference
SUBROUTINE                ;start a subroutine
CALL                      ;invoke a subroutine name
                          ;by SUBROUTINE statement
RETURN                    ;return from subroutine
EQUIVALENCE               ;assign same storage to
                          ;variables
```

Functions

```
ABS (XXX)                 ;Absolute Value (real numbers)
IABS (XXX)                ;Absolute Value (integer numbers)
FLOAT (XXX)               ;Convert XXX from integer to real
IFIX (XXX)                ;Convert XXX from real to integer
IREM (XXX)                ;Remainder of last integer division
EXP (XXX)                 ;e^X
IRDSW                     ;Read console switch register
```

```
NAME                      DESCRIPTION

ALOG                      ;Natural log
SIN, COS, TAN             ;Sine, Cosine, Tangent
ATAN                      ;Arctangent
SQRT                      ;Square root
```

```
Operators and Special Characters
   +,-,/,*                    ;Add, subtract, divide, multiply
   **                         ;Exponentiation
   =                          ;Assignment

Format Specifications (in FORMAT statement)

   I                          ;Integer
   E                          ;Floating point with exponent
   F                          ;Floating point with no
                              ;exponent
   A                          ;Alphanumeric field
   H                          ;Hollerith
```

Source: *Programming Lanugages, PDP-8 Family Computers,* Digital Equipment Corporation, 1970.

The sample Fortran program below compares two character strings. Program is written in the FORT/80 dialect. Courtesy of Realistic Control Corporation.

```
              INTEGER*1 FUNCTION EQUAL
             X(STRING1, STRING2)
C             DECLARE BOTH ARRAY PARAMETERS DYNAMICALLY
              INTEGER*1 STRING1(1), STRING2(1)
              EQUAL = .FALSE.
              I = Ø
C             VERIFY THAT THE STRINGS AGREE
C             UNTIL AN ETX (ASCII Ø3) IS FOUND
1ØØ           IF (STRING1(I=I+1) .EQ. Ø3H)
             XGO TO 2ØØ
              IF (STRING2(I     ) .EQ. Ø3H)
             XGO TO 3ØØ
              IF (STRING1(I) .NE. STRING2(I) RETURN
              GO TO 1ØØ
C             VERIFY THAT THE REMAINDER OF STRING2
C             IS BLANK
2ØØ           IF (STRING2(I) .EQ. Ø3H) GO TO 4ØØ
              IF (STRING2(I) .NE. ' ') RETURN
              I = I + 1
              GO TO 2ØØ
C             VERIFY THAT THE REMAINDER OF STRING
```

```
C          IS BLANK
3ØØ        IF (STRING1(I) .EQ. Ø3H) GO TO 4ØØ
           IF (STRING1(I) .NE. ' ') RETURN
           I = I + 1
           GO TO 3ØØ
C          THE STRINGS ARE EQUAL
4ØØ        EQUAL = .TRUE.
           RETURN
           END
```

A Summary of the PL/M language

Purpose: PL/M is a language designed to replace Assembler language programming yet retain complete control of machine level functions such as interrupts and Input/Output. It is usually compiled.
Data Types: PL/M supports BYTE (8 bit) and ADDRESS (16 bit) data types. In addition, the compiler can reference the address of a variable (pointer).
Data Structures: PL/M supports both scalar and single dimension arrays. Arrays with length zero are ghost arrays which may not be used to reference memory not specifically assigned to it.

Commands:

NAME	DESCRIPTION
IF/THEN/ELSE	;Permits conditional execution
DO	;Begin block; provide looping ;capability
DO CASE	;Select one of several suc- ;ceeding statements
PROCEDURE	;Begin program or subprogram; ;name parameters
END	;End DO, PROCEDURE or INTER- ;RUPT blocks
DECLARE	;Declare variables or arrays, ;and their attributes
GOTO	;Unconditional branch
CALL	;Invoke a subroutine
RETURN	;Return from a subroutine
HALT	;Machine stop
ENABLE	;Enable interrupts
DISABLE	;Disable interrupts
EOF	;End of compiler input

Functions

MOD	;Remainder after division
PLUS	;Add with carry
MINUS	;Subtract with borrow
INPUT (XXX)	;Read byte from port XXX
OUTPUT (XXX)	;Write byte to port XXX
LOW	;Convert low order 8 bits of ;ADDRESS to BYTE value
HIGH	;Convert high order 8 bits ;of ADDRESS to BYTE value
DOUBLE	;Convert a BYTE value to an ;ADDRESS value
ROL	;Rotate BYTE value to left
ROR	;Rotate BYTE value to right
SCL	;Rotate BYTE value left through ;carry
SHL	;Shift BYTE value left logically
SHR	;Shift BYTE value right logically
TIME (XXX)	;Delay program for XXX ;100 microseconds

Source: *8008 and 8080 PL/M Programming Manual,* Intel Corporation, 1975

The sample PL/M* program below compares two character strings for equality. It is similar to the FORTRAN example except that the PL/M program does not check for different length strings. Source: ***PL/M Programming Manual,*** **Intel Corp.** *Trademark, Intel Corp.

```
EQUAL: PROCEDURE (PTR1, PTR2) BYTE;
/*
    THIS PROCEDURE COMPARES CHARACTER STRINGS
    FOR EQUALITY. THE FINAL BYTE OF EACH STRING
    MUST BE OFFH.
    IF THE STRINGS ARE EQUAL, OFFH(=TRUE) IS
    RETURNED, OTHERWISE 0(=FALSE) IS RETURNED.
                                              */
```

GENIE

```
    DECLARE (PTR1, PTR2) ADDRESS
    DECLARE (STRING1 BASED PTR1,
             STRING2 BASED PTR2) BYTE;
    DECLARE I ADDRESS, (J1, J2) BYTE;
    J1, J2, I = 0;
    DO WHILE J1=J2;
     IF J1=OFFH THEN RETURN OFFH;
     J1 = STRING1(I);
     J2 = STRING2(I);
     I=I+1;
    END;
    RETURN 0;
  END EQUAL;
  EOF
```

A Summary of the MUMPS Language

Purpose: MUMPS is an interpretive, interactive data management language suited for information systems and text processing.

Data Types: Formally, MUMPS has only one data type — the variable length character string. However, special cases of this data type are integer, floating point, and logical. MUMPS handles conversions between all these automatically.

Data Structures: MUMPS supports scalars and sparse subscripted arrays to any number. Data types of elements within arrays may be intermingled, and subscripts need neither be contiguous nor dimensioned before use. Data may be stored in main memory (local array) or on a direct access device (global array). A global array is treated exactly like a local array, only references to it are preceded by a circumflex (^).

```
Commands:

   NAME                DESCRIPTION

   BREAK               ;Suspend operation until pro-
                       ;grammer signals to resume execution
   CLOSE XXX           ;Release device XXX from ownership
                       ;(see OPEN)
   DO XXX              ;Generalized subroutine reference
   DO ^ XXX            ;Reference XXX names part of current
                       ;routine; XXX names routine on disk
                       ;to be loaded
   ELSE                ;Permits conditional execution
                       ;after IF statement
   FOR XXX             ;Loop through following code according
                       ;to arguments XXX. Arguments may con-
                       ;tain numbers, character strings, or
                       ;loop boundaries.
   GOTO XXX            ;Transfer control to XXX. If
   GOTO ^ XXX          ;^XXX, load program XXX, then transfer
                       ;control.
   HALT                ;Terminate execution of program
   HANG XXX            ;Suspend execution XXX seconds
   IF                  ;Permits condition execution
   KILL XXX            ;Delete variable XXX from partition
   KILL  XXX           ;(or disk if preceded by ^). If
                       ;argument enclosed in parentheses,
                       ;delete all but XXX.
   LOCK XXX            ;Claim exclusive use of resources
                       ;XXX. If any other programs have
                       ;locked resources named by XXX,
                       ;force this program to wait until
                       ;they are freed with a lock state-
                       ;ment with no arguments.
   OPEN XXX            ;Request exclusive use of device.
   QUIT                ;Terminate execution of DO, FOR, or
                       ;Xecute Command.
   READ XXX            ;Read data from last device specified
                       ;by a USE command. XXX can contain
                       ;extensive terminal control features.
   SET XXX=YYY, . .    ;Assign value of right-hand expression
                       ;YYY to left-hand variable XXX. Note
                       ;expressions can be data bases.
   USE XXX             ;Use device XXX for all future READ
                       ;and WRITE commands.
   VIEW XXX            ;Read or write data into absolute
                       ;locations in memory.
   WRITE XXX           ;Write data XXX to device defined by
```

XIMEDIA PRESENTS

The SOROC IQ120

CURSOR CONTROL. Forespace, backspace, up, down, new line, return, home, tab, PLUS ABSOLUTE CURSOR ADDRESSING.

TRANSMISSION MODES. Conversation (half and full Duplex) PLUS BLOCK MODE — transmit a page at a time.

FIELD PROTECTION. Any part of the display can be "protected" to prevent overtyping. Protected fields are displayed at reduced intensity.

EDITING. Clear screen, typeover, absolute cursor addressing, erase to end of page, erase to end of line, erase to end of field.

DISPLAY FORMAT. 24 lines by 80 characters (1,920 characters).

CHARACTER SET. 96 characters total. Upper and lower case ASCII.

KEYBOARD. 73 keys including numeric key pad.

REPEAT KEY. 15 cps repeat action.

DATA RATES. Thumbwheel selectable from 75 to 19,200 baud.

SCREEN. 12 inch rectangular CRT — P4 phosphor.

SPECIAL INTRODUCTORY PRICING

Kit $ 995.00 **Assembled $ 1,295.00**

(Price includes block mode, lower case and 24 line options.)

Specials of the Month

North Star **Micro Disk** with power supply and cabinet ..... Kit — $699, Assembled — $799

IMSAI I-8080 with **TDL** ZPU ...... Kit — $825, Assembled — $975

Digital Systems FDS Disk Drive with **CP/M** Software (assembled only) . Single — $1,750, Dual — $2,350

Mountain Hardware **PROROM** ..... Kit — $145, Assembled — $210

Vector Graphic 8K RAM .... Kit — $235, Assembled — $285

NOW WE'RE TOLL FREE

800-227-4440

(in California, Hawaii, and Alaska, call collect: 415/566-7472)

XIMEDIA OFFERS A FULL RANGE OF PRODUCTS FOR THE PERSONAL COMPUTER ENTHUSIAST AND THE SMALL SYSTEM DESIGNER. LET US QUOTE ON ALL YOUR HARDWARE AND SOFTWARE NEEDS.

OUR RETAIL STORE — THE COMPUTERIST™ — IS NOW OPEN IN SAN FRANCISCO. CALL US FOR DIRECTIONS.

XIMEDIA

1290 24th Avenue, San Francisco, CA 94122
(415) 566-7472

COD orders freight collect. Orders with payment shipped prepaid. California residents add 6% sales tax. Please allow 3 weeks for delivery.

CIRCLE 35

GENIE

```
                        ;last use command. XXX can contain
                        ;terminal control features.
  Xecute XXX            ;Treat argument XXX as a line of
                        ;MUMPS code.
  Z                     ;Reserved for implementation -
                        ;specific commands. (All nonstandard
                        ;commands must start with Z).

Functions

  NAME                  DESCRIPTION

  $ASCII (XXX)          ;Convert argument from character
                        ;string to its decimal equivalent.
  $CHAR (XXX)           ;Convert argument from decimal number
                        ;to ASCII character.
  $DATA (XXX)           ;Return information describing variable
                        ;XXX's status.
  $EXTRACT (XXX)        ;Extract character string from another.
  $FIND (XXX)           ;Find a string within another.
  $JUSTIFY (XXX)        ;Right justify and round a number
                        ;into a string.
  $LENGTH (XXX)         ;Return the length of a string.
  $NEXT (XXX)           ;Return the subscript number of the
                        ;next sequential element in array XXX.
  $PIECE (XXX)          ;Return a "piece" of a character string.
  $RANDOM (XXX)         ;Return a random number between 0 thru
                        ;XXX-1.
  $SELECT (XXX)         ;Execute one of a series of commands,
                        ;based on a series of truth values.
  $TEXT (XXX)           ;Return the program line defined by
                        ;XXX as a character string.
  $VIEW (XXX)           ;Read or write absolute memory addresses.
  $Z (XXX)              ;Implementation - specific functions·
                        ;all non-standard functions must begin
                        ;with Z.

Special Variables

  $Horolog              ;Returns the time and date.
  $IO                   ;Returns the current device number.
  $STORAGE              ;Returns the number of bytes of storage
                        ;available to program.
  $TEST                 ;A special flag containing value of
                        ;last computed truth value.
  $X                    ;Horizontal position of cursor on
                        ;current device.
  $Y                    ;Vertical position of cursor on
                        ;current device.

Operators

  &                     ;Logical AND
  !                     ;Logical OR
  '                     ;Logical NOT
  +,-,/,*               ;Add, subtract, divide, multiply
  \                     ;Integer division
  #                     ;Modulo
  < , >                 ;Less than, greater than
  _                     ;String concatenation
  [                     ;String search
  X ] Y                 ;String ordering - does string X
                        ;follow string Y?
  X ? Y                 ;Pattern match - does string X
                        ;match pattern Y?

  Pattern match codes:

  A                     ;All 26 upper and lower alphabetics
  C                     ;Control characters
  E                     ;Entire character set
  L                     ;Lower case alphabetic
  N                     ;Numerics (0-9)
  U                     ;Upper case alphabetics only
  @                     ;Indirection. Any argument preceded
                        ;by an @ refers to the name of a
                        ;variable to be used as the argument.
  ^                     ;Global reference. Any variable pre-
                        ;ceded by an indicator that it resides
                        ;on disk.
```

Source: *Standard MUMPS Pocket Guide,* MUMPS User's Group, 700 South Euclid Ave., St. Louis Mo 63110.

The following sample MUMPS program averages student grades, then stores the results on a data base for future use. MUMPS is a language oriented to data manipulation.

GET THE BEST FROM... SEALS

ORDER FORM [Write in each box the quantity of each part required]

KIT		ASSEMBLED	KIT		ASSEMBLED	KIT		ASSEMBLED
☐	4KROM	☐	☐	WWC	☐	☐	68EXT-L	☐
☐	8KSC-Z	☐	☐	88 EXT	☐	☐	68WWC	☐
☐	68KSC	☐	☐	68EXT-S	☐	☐	BBUC	☐
☐	8KSC	☐						

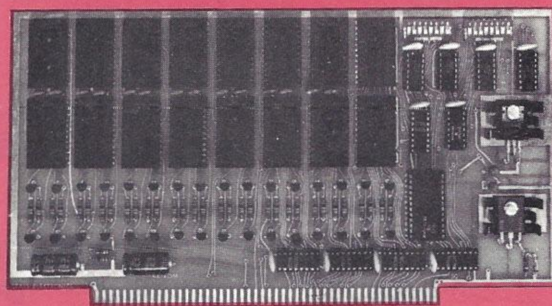
☐ ASSEMBLY AND OPERATING MANUAL $4.00

NAME ______________________
PLEASE PRINT OR TYPE
ADDRESS ______________________
CITY ______________ STATE ________ ZIP ________

SEND CHECK • MONEY ORDER • COD'S ACCEPTED • CREDIT CARDS

SEALS ELECTRONICS, INC
P.O. BOX 11651
KNOXVILLE, TN 37919
MOST ORDERS SHIPPED WITHIN 10 WORKING DAYS

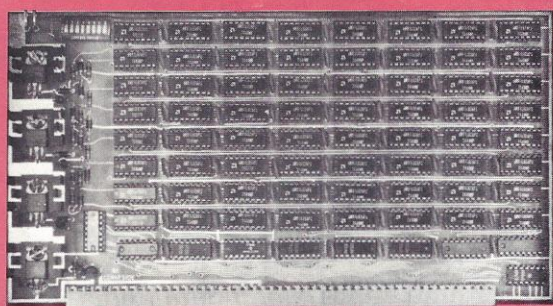
COPY ACCOUNT NUMBER FROM YOUR BANK AMERICARD
BANKAMERICARD
MY CARD EXPIRES
COPY ACCOUNT NUMBER FROM YOUR MASTER CHARGE
master charge
Copy number above your name on Mastercharge
MY CARD EXPIRES



4KROM [READ ONLY MEMORY]

- Will accept up to 16 ea. 1702-A or 5203 EProm Providing up to 4096 words of non-volatile memory for Boot Loads to Complete Programs.
- Programming Available at Factory for $3.00 per EProm when accompanied by binary formated tape.
- Each 1702-A has its own Vgg clocked for Low Power Consumption.
- Will work with the weakest power supply based S-100 buss computer.
- Switched Selected Address in 4K Blocks.
- Switch selected wait states so that even the slowest 1702-A can work in your system. 0-8 wait states.
- Solder Masked on both sides of PC Board.
- Component Screened on Component Side of PC Board.

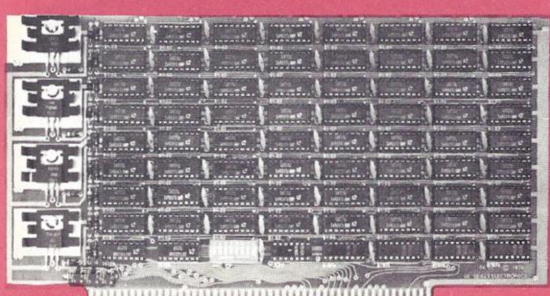
Kit Price: . . $119.00 **Assembled Price: . . .$179.00**



68KSC [STATIC MEMORY CARD FOR SWTPC 6800]

- 8192 Words of Static Memory
- Access Time: 500 nsec. (250 nsec on request)
- Memory Chip 91L02 APC or 2102AL-4
- Battery Standby
- Address Selected 8 Ea. SPST Dip Switch
- Low Power
- All Lines Buffered.
- All IC's with sockets
- Solder Masked on both sides of PC Board.

Kit Price: . . $269.00 **Assembled Price: . . .$359.00**



8KSC-Z [STATIC MEMORY CARD]

- ALTAIR®, IMSAI®, and S-100 buss compatible.
- Access Time: 250 nsec max.
- Ziloq Speed Compatible up to 4 mhz.
- Memory Chip: 2102LHPC or 2102AL-2
- Battery Standby: >1.5 to 4 volts<
- Address Select: 8 ea. Spst. Dip Switch.
- Wait States: None
- Current Reg.: Less than 200 ma per 1K
- All Address, Control, and Data out lines fully buffered.
- All IC's supplied with IC Sockets
- Solder Masked on Both Front and Back of P.C. Board.

Kit Price: . . $295.00 **Assembled Price: . . .$395.00**

OTHER SEALS ELECTRONIC PARTS AVAILABLE

Item No.	Description	Kit Price	Assembled Price
8KSC	8K Static Memory Card 500 nsec	$269.00	$369.00
WWC	Wire Wrap Card	$ 37.50	$ 47.50
88 EXT	88 Extender Card	$ 29.00	$ 38.00
68 Ext-S	Extender Card (Small) SWTPC 6800 Compatible Products	$ 19.00	$ 25.00
68 Ext-L	Extender Card (Large) SWTPC 6800 Compatible Products	$ 29.00	$ 39.00
68 WWC	Wire Wrap Card SWTPC 6800 Compatible Products	$ 35.00	$ 45.00
BBUC	Battery Back-Up Card	$ 55.00	$ 68.00

SEALS® ELECTRONICS, INC.

ORDER DELIVERY: FROM STOCK TO 10 DAYS TELEX NO. 55-7444 TELEPHONE NO. 615/693-8655

CIRCLE 38

GENIE

```
DEMO ; THIS PROGRAM TAKES AVERAGES OF STUDENT GRADES
     ; AND STORES THEM ON A DATA BASE
     SET TOTAL=0,NUM=0
     WRITE !"ENTER CARRIAGE RETURN WHEN DONE WITH GRADES"
LOOP READ !"ENTER STUDENT GRADE",GRADE
     IF GRADE="" GOTO FINE
     SET TOTAL=TOTAL+1,NUM=NUM+GRADE GOTO LOOP
FINE READ !"ENTER CLASS NUMBER",CLASS
     SET GRADES(CLASS)=TOTAL/NUM
```

A Summary of the FOCAL Language

Purpose: FOCAL (Formulating On-Line Calculations in Algebraic Language) is an interpretive calculation language.

Data Types: FOCAL supports floating point numbers.

Data Structure: Scalars or single dimension arrays.

Commands:

NAME	DESCRIPTION
ASK XXX	;input variables from keyboard
COMMENT	;program comments - ignored by ;interpreter
DO XXX	;subroutine reference allows ;looping according to XXX
ERASE XXX	;erases program lines XXX
FOR XXX	;loop through following code ;according to arguments XXX
GO	;execute lowest number line in ;program
GOTO XXX	;starts executing at line XXX
GO? or GOTO?	;same as GO and GOTO above, only ;activate trace feature
IF	;permits conditional execution
Library CALL	;calls stored program from disk
Library DELETE	;deletes program from disk
Library LIST	;list programs on disk
Library SAVE	;saves program on disk
QUIT	;return control to user
RETURN	;terminates DO routines
SET XXX=YYY	;assign value of right hand ;expression to left hand variable
TYPE XXX	;evaluate expression XXX and type ;result on console device
WRITE XXX	;list program lines defined by XXX

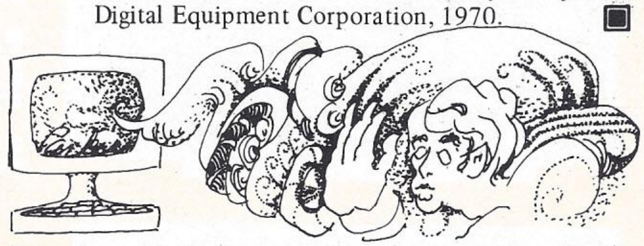
Functions

FSQT	;square root
FABS	;absolute value
FSGN	;sign of expression
FITR	;integer part of expression
FRAN	;random number generator
FEXP	;number base to power
FSIN	;sine
FCOS	;cosine
FATN	;arctangent
FLOG	;log
FNEW	;user function
FCOM	;storage function

Operators and Special Characters

+,-,/,*	;add subtract, divide, multiply
^	;exponentiation
%	;output format delimiter
!	;carriage return, line feed
#	;carriage return
$	;type symbol table contents

Source: *Programming Language, PDP-8 Family Computers,* Digital Equipment Corporation, 1970. ■



Featured Artist

Manuel Barbadillo uses the computer not as a drawing instrument, but as a planning instrument, a fast experimental device that lets him try many complex ideas roughly before choosing one that satisfies him for further development. He comments that he prefers a line printer – with asterisks roughly filling the spaces – to a plotter. Typically, he renders the final work by hand.

From his base in Torremolinos, Spain, Barbadillo has provided a statement that will help the viewer who seeks knowledge, as well as pleasure, from the handsome work shown here. Artists' explanations, like artists' art, must speak for themselves. Says he:

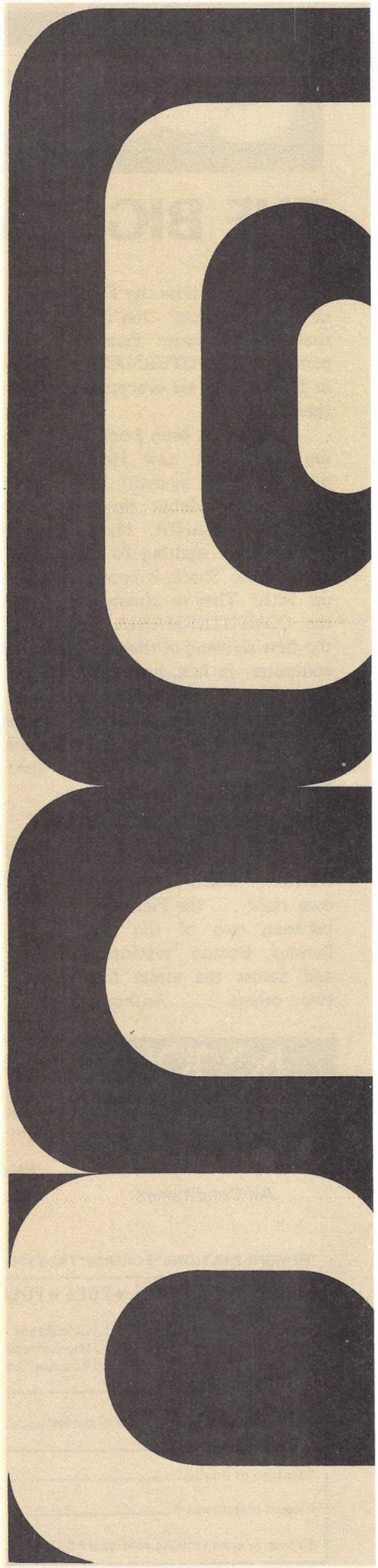
"My painting is a pluri-directional writing. Its alphabet is a set of basic signs whose definitions in a square are absolutely objective. (I call them modules.) These signs, like in music, have no fixed or conventional meaning by themselves; they acquire it through their structural relationships when articulated in a grid.

"They consist of areas of two opposite colours. Their outlines have straight and curved parts, all of which are generated by the same measure.

"The spatial development of a painting, by addition of equal straight segments, of curves with the same radius, or of the former with the latter defining the contour of white and black "figures" is thus rhythmic, with – speaking metaphorically – a beat or a pulse accounting for the integration of the parts into a whole.

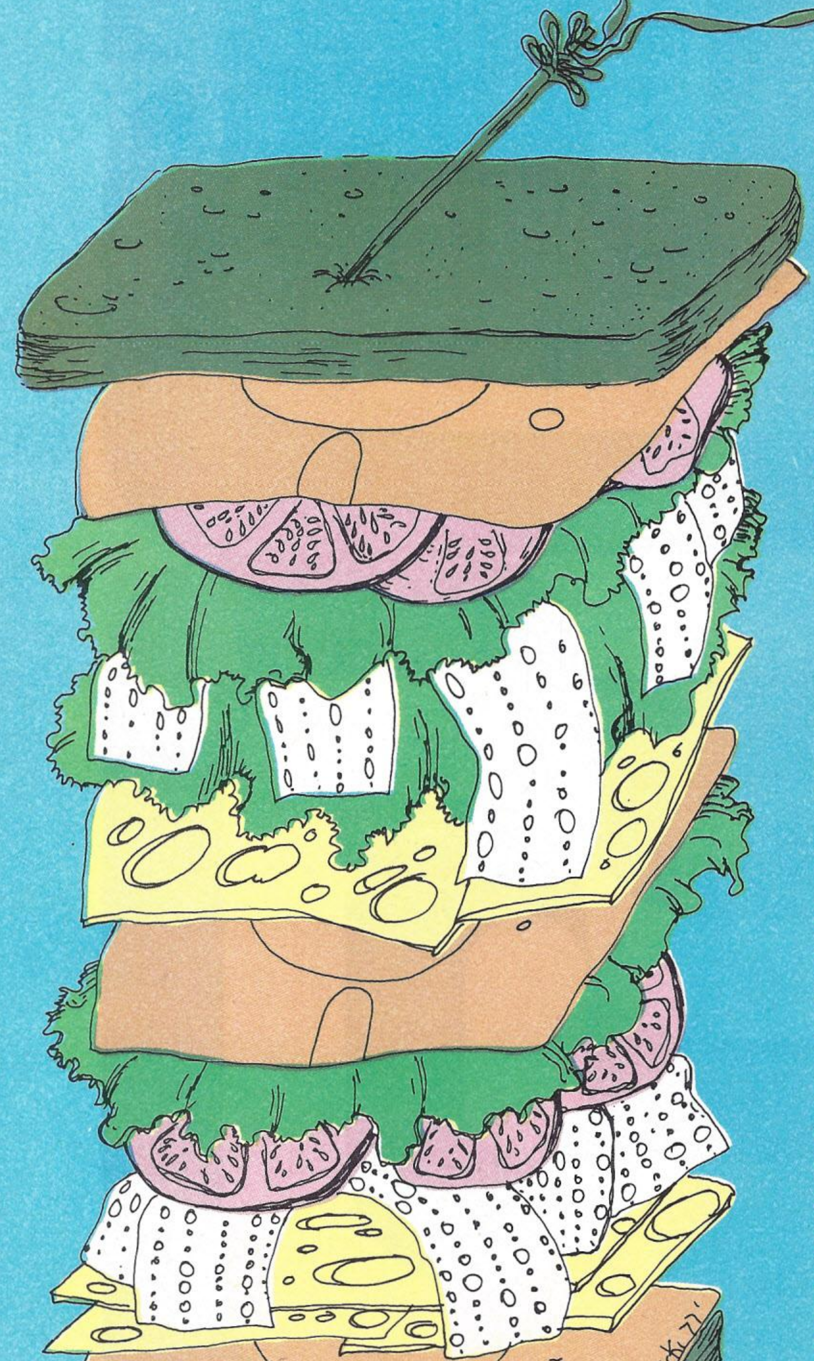
"My method for making a picture is mainly intuitive. I just try to express myself with the modules like poets do with words or musicians with notes, creating a rhythmic structure. But with the limitations inherent in the module's designs and my use of a grid as a track, plus a certain amount of systematization in my "doodling."

"These later features in my work drove me naturally to the computer, which I have used as well to know more about my work – as a help for analysis – as to get visual stimuli in the search for "meaningful" configurations. Computer analysis has revealed hidden mathematical rules in compositions that had been subjectively – emotionally – considered "meaningful." "



CLUB ROUNDUP

by Louise Garcia



Connecticut SCCS.. "Buss"

Charles Floto is raring to go on a users group organizing for the exchange of information on use of Heathkits by computer hobbyists. The group will not in any way be connected with Heath Company. Buss will provide early news on the Heathkit computers as it's available – contact Charles Floto, 267 Willow Street, New Haven, Connecticut 06511.

HP-65 Users Club.. "65 Notes"

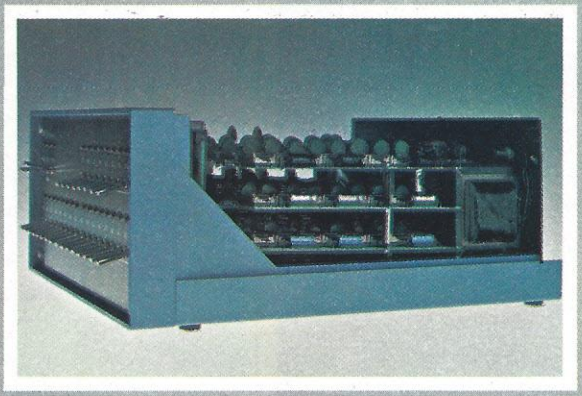
Are you a calculator user? There's a club just for you. The HP-65 Users Club is organized to save effort and to increase the use of the calculator by sharing applications knowledge. Membership is $12.00 and it buys you the monthly newsletter called 65 NOTES, a membership list containing special interests of members and an "in" with the program swapping system.

"65 NOTES" reports that "the members of the HP-65 Users Club are the most informed calculator users in the world; virtually *every* useful or spectacular calculator technique or discovery has been reported in 65 NOTES as discussed elsewhere, or, as is the usual case, reported in detail by the originator . . . ."

An informative and entertaining piece called "Pinball Wizzard" by Craig A. Pearce (#311) is a nice treatment of standard pinball with a program that's designed to be left running at all times – all input from the user occurring during the active-keyboard-during-pause feature of the HP-67/97. Pearce cautions: "Be prepared". This pinball makes it necessary for the user to participate during the play . . . response and use of the "flippers" has to be immediate – otherwise – oops, there go the bonus points.

Powerful in computing muscle, yet small in physical size, the Altair™ 680b offers many special features at an affordable price. Based on the 6800 microprocessor, the 680b comes with 1K of static RAM, Serial I/O port, PROM monitor and provisions for 1K of PROM as standard components. It's good thinking, when you're interested in making a modest investment on a highly reliable computer, to consider the Altair 680b.

Our PROM monitor eliminates the necessity for toggling front panel switches to load bootstraps or manipulate memory contents. Only a terminal and programming language are required for complete system operation. With Altair System software—Altair 680 BASIC, assembler and text editor—you may begin problem solving immediately with ease.



By adding the 680b-MB Expander card, many options are currently available:

*16K Static Memory Board—Increase your system memory with 16K bytes of fast access (215 ns), low power (5 watts per board) static RAM. 680 BASIC and assembler/text editor are included free with purchase.

*Process Control Interface—A PC card that uses optically isolated inputs and relay outputs that transmit sensory information to and control signals from the computer. A diverse world of control applications is opened up with the Altair 680b-PCI.

*Universal Input/Output Board—If your I/O needs exceed the serial port already on the main board, augment your I/O channels with the 680b-UI/O. By implementing the optional serial port and two parallel ports, you can simultaneously interface to four terminals.

*New Addition—Kansas City Audio Cassette Interface—Use the 680b-KCACR to interface your Altair 680b with an audio cassette recorder for inexpensive mass storage of programming languages, programs and data.

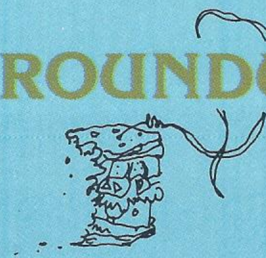
Available in either full front panel or turnkey models, the Altair 680b presents many computing capabilities at a low cost—without skimping on performance. See it today at your local Altair Computer Center or contact the factory for further details.

mits

Good Thinking.

2450 Alamo S.E. Albuquerque, New Mexico 87106
dealer inquiries invited.

CLUB ROUNDUP



John Kennedy (#918) briefly describes his recent contribution of CHESS CLOCK. Although it's not *exactly* a chess clock it can be modified to that. It's ideal for one-minute-per-move chess plays and other games.

Twin Cities Technical Hobbyist..

Working with the theme "Computing is Fun – with Something for Everyone", Minnesota Computer Society members have begun to chart their course for the year's events. Some plans include sponsoring, co-sponsoring classes of interest to members, providing workshops at society meetings, beginning hardware/software group purchases, establishing special interest groups. If you're interested or have suggestions and ideas for future plans for the MCS, contact Minnesota Computer Society, Box 35317, Minneapolis, Minnesota 55435.

Rochester Area Microcomputer Society.. "Memory Pages"

A call out to anyone interested in forming a group to find a cheap floppy system – please contact Dave Noderer at 716-288-3248, evenings and 716-482-5000, x 144, days.

San Diego Computer Society.. "Personal Systems"

A report entitled "The Micro Tower . . . Comparing Instruction Sheets" examines a few of the criteria for evaluating instruction sets supplied by computer manufacturers, and shows how some microprocessors stand up to those criteria. The article indicates that almost all computer manufacturers claim their machine's instruction set is the easiest to learn and use. By working with a list of "factors" the reader determines the power behind the instruction set as well as

CLUB ROUNDUP

the learning ease. Additionally, the author lists some of the more common restrictions found on instruction sets and compares them to factors like consistency, simplicity and straightforwardness, which make the programmer's tasks much easier. Brief article, but mighty informative.

The San Diego Computer Society is an organization whose purpose is to provide its members and the general public with a usable source of computer-related technology info. Dues are $10.00 – contact the Society at P.O. Box 9988, San Diego 92109.

Washington Amateur Computer Society.. "The Journal"

The first electronic journal! The Journal is readily available as a free access text file on the DECSystem-10 of Catholic University. No password – no account number! Needed is a modem and a terminal. Set your modem (acoustic coupler) and terminal to full duplex, no parity. Dial 202-635-5710 if you have a 110 Baud terminal (such as a standard teletype) or 202-635-5730 for a 300 baud terminal. You pay for the phone call. When the computer responds with its audio signal, type a "Control – C". The computer will respond with an "Up-arrow and a C". Type an "I" followed by a carriage return. The computer will respond with a brief message and a "period". Type "Help WACS" and a carriage return to get the Journal. You can interrupt the output at any time by typing a "Control O" (the letter O, not the number 0). When you're through, just hang up.

Of course the Journal is still available in printed form (with a print-out of Snoopy as a bonus).

The directors and editors of the WACS Journal firmly believe that this electronic system is a major breakthrough in newsletters and hope it's a first of many to come from other organizations. They say that hobbyists need access to quick, efficient and up-to-date communications media to help hobbyists grow and exchange ideas – and here 'tis.

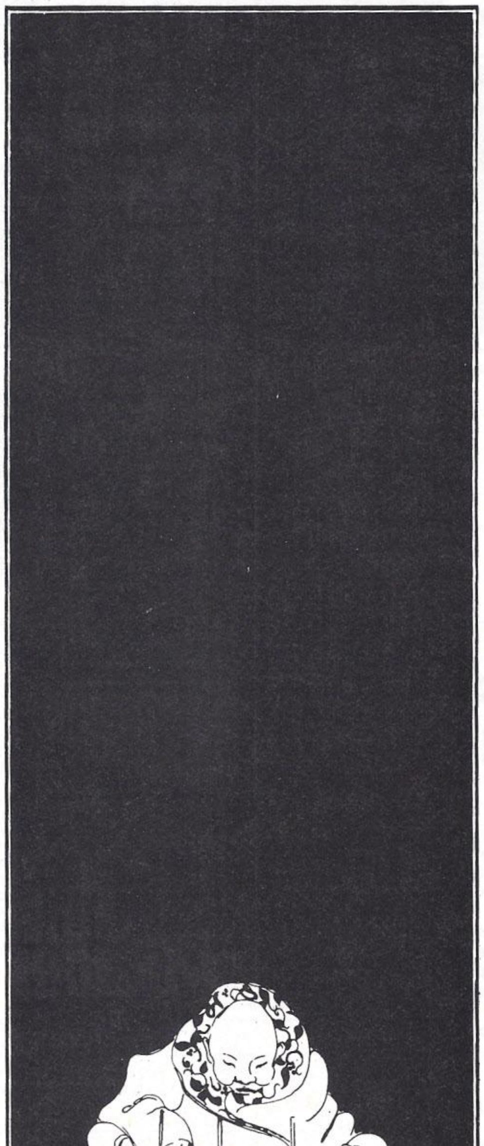
WACS is an affiliate of the Chesapeake Microcomputer Club. Contact Bob Jones, Editor, 4201 Massachusetts Ave., Apt. 168, Washington, D.C., 20016 for more information.

REMEMBER: Put PERSONAL COMPUTING on *your* mailing list!

Programming loops are the key

Get Rolling With Loops

By Steve A. Hughes, Joe Celko and Elizabeth M. Hughes



An important characteristic of computers – one that has gone far to win them their place in business and industry – is their ability, when properly programmed, to perform repetitive or complex operations with speed and accuracy and without succumbing to boredom. Rarely is a computer program written with the intent to use it only once and even more rare is the program which doesn't involve, somewhere in its structure, a repeated series of operations. For this reason, *loops*, the programming structure for efficient handling of repetition, are one of the most commonly used of all programming devices.

Do It With a Loop

Whenever it is necessary to perform an operation more than once, one can choose between repeating the code which performs it or using a loop. Since loops require less code (hence, less storage) and since it is not necessary when writing a loop to know exactly how many times the operation it performs will be repeated, loops are almost invariably the better choice (Fig. 1). By decreasing the amount of code required, they reduce the likelihood of error; and since, with a loop, it is possible to determine whether an operation is needed before it is performed, computer run-time can sometimes be reduced. In addition, loops can be used for non-computational tasks, such as timing, waiting and data manipulation. Loops, then, are one of the most potent weapons in the programming arsenal.

Although loops vary considerably according to the needs of the program and the limitations of the programming language, it is generally possible to distinguish three parts: the BODY, which performs the repeated operation; the TEST FOR EXIT CONDITION, which determines whether or not the loop's task has been completed; and the RETURN instruction, which returns control to the beginning of the BODY when the TEST indicates that it is not yet time to exit from the loop. Perhaps the best analogy for a loop is the "all-you-can-eat" buffet in a restaurant. The BODY of the loop is the operation of filling a plate and consuming the food which has been collected, and the TEST occurs when the patron decides whether to go back for more. If he decides to eat more, he RETURNS to the buffet line and repeats the operation; if he decides otherwise, he EXITS from the restaurant. That, essentially, goes on in a loop.

Applications for Loops

Loops can be used for many purposes. If, for example, it is desirable to keep the computer running when it is not actually running a program (to save volatile memory, for instance), a pause condition can be established by using a JUMP TO PRESENT INSTRUCTION command. This effective-

to unlocking computer power

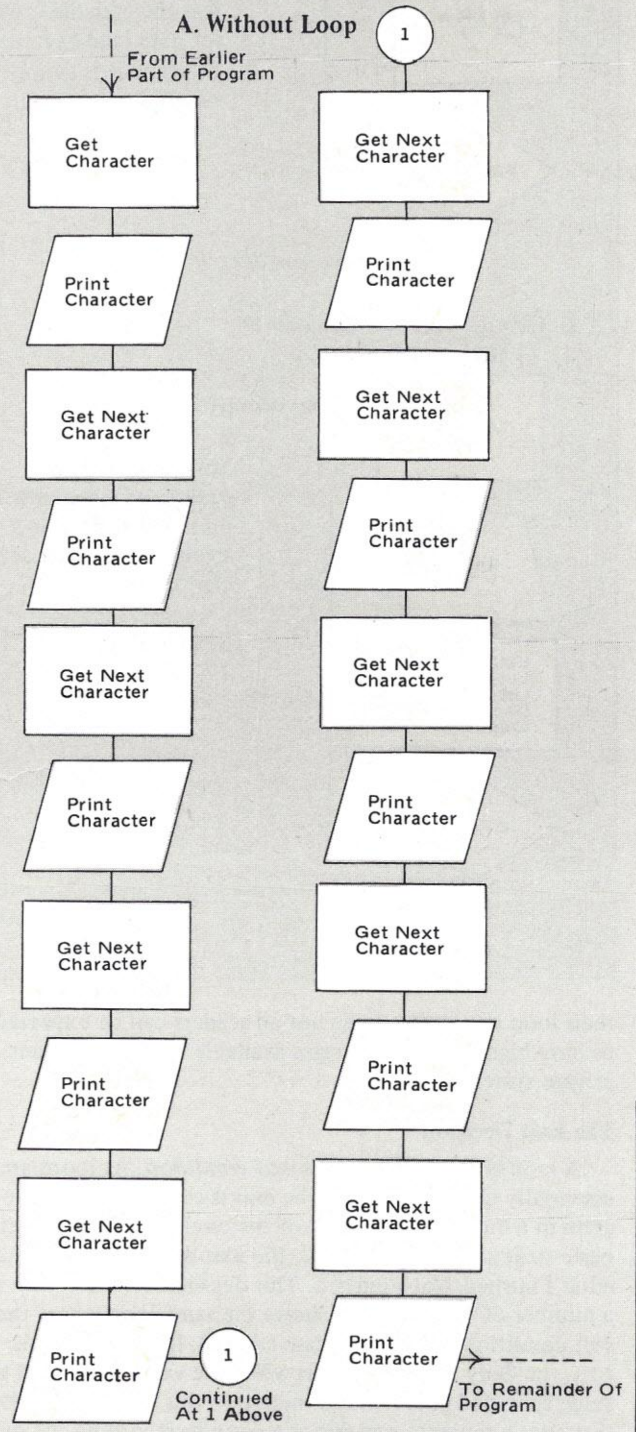
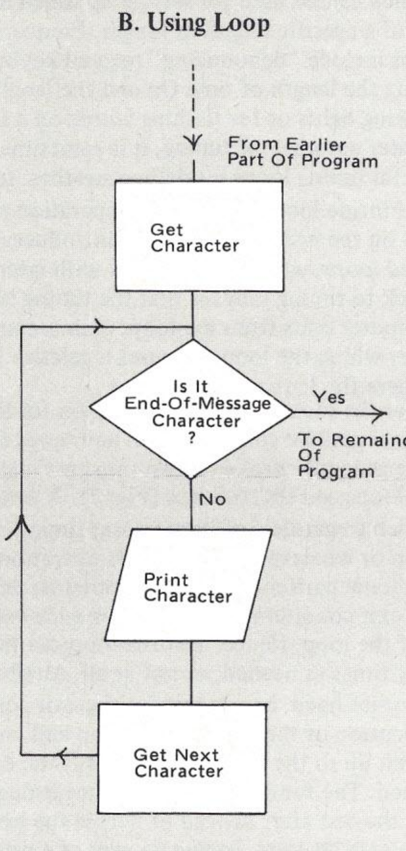


Fig 1 – Loops permit shorter programs than are possible using repeated code. Example: flowchart of a program segment to output ten characters.



Note also that, as flowcharted, B does not require any maximum or minimum message length.

ly sets the computer to twiddling its thumbs until it is given an INTERRUPT and a fresh command. *Waiting loops* of a somewhat more complicated form can be included in programs to cause the delay while waiting for some particular condition, such as human intervention, to occur; and they are frequently useful in I/O operations. Another common use for loops is data manipulation. If, for example, it is desirable to move a block of data, one word at a time, from one area in memory to another, the task can be accomplished easily by the use of a loop.

Timing loops are a third common application. They re-

Fig 2 - WHILE-Loops and UNTIL-Loops

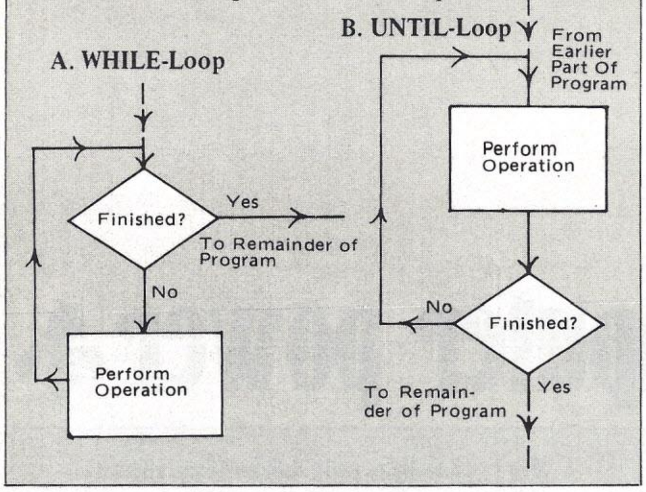
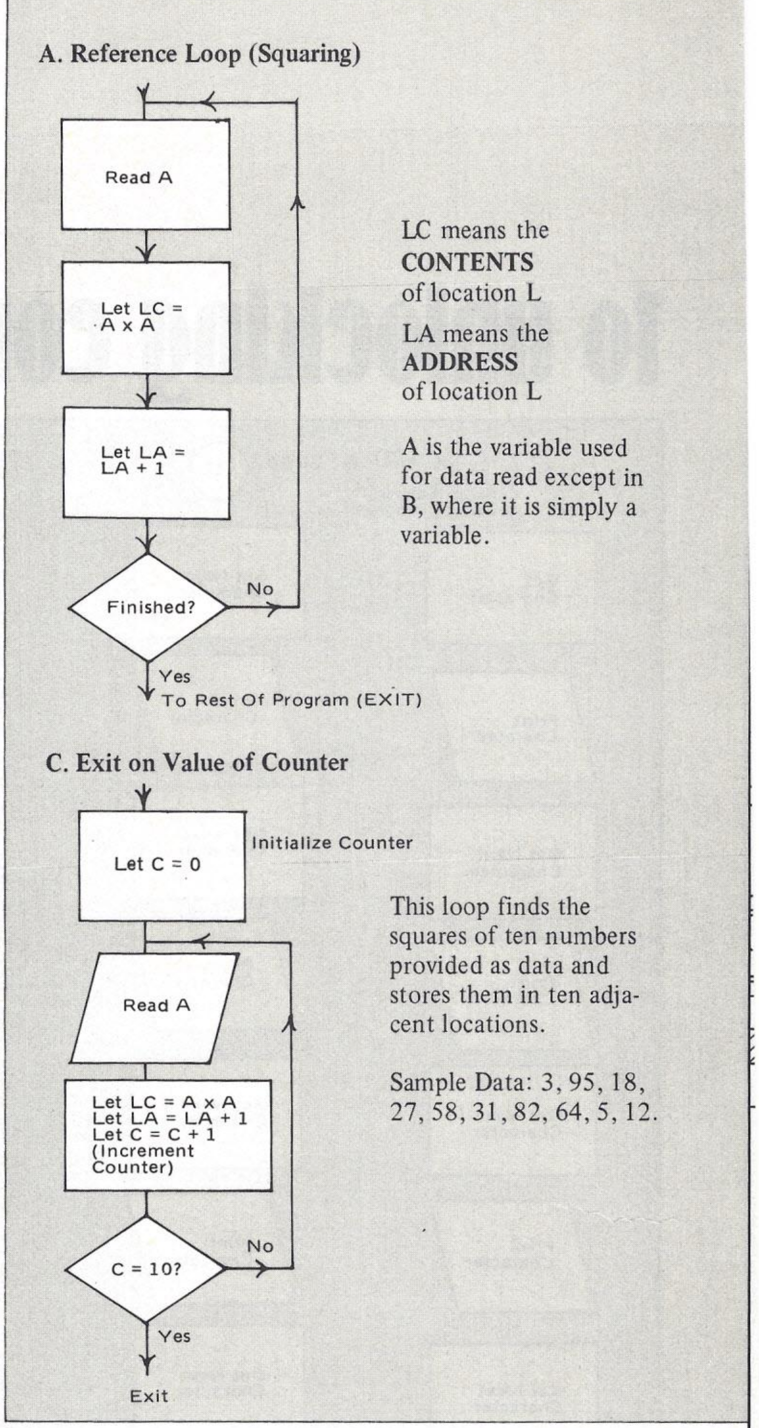


Fig 3 - Some Ways of Exiting From Loops.



semble waiting loops, except that the amount of time required for the execution of one pass through a timing loop is planned and known. This provides a known measure of time which can be used for setting up timed delays or timed actions of a specific length or longer. Frequent uses of timing loops include "debouncing" manual keyboards and determining the length of time On and the length of time Off for blinking lights or for flashing words on a CRT display. For greater accuracy of timing, it is sometimes desirable to put several timing loops inside one another, so that each time the inside loop completes its operation it increments a counter on the next loop; but this introduces the question of *nested loops*, which will be dealt with later. The only drawback to timing loops is that the timing is lost whenever the computer exits from the loop; each second, minute or whatever which the loop measures is relative to the point in time where the loop was entered.

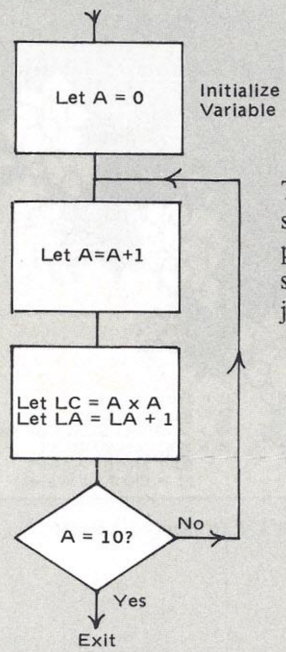
By far the most common application for loops, however, is for repetitive computations, and loops for this purpose are generally broken down into two major types: WHILE-loops and UNTIL-loops (Fig. 2). A WHILE-loop is one which (regardless of its purpose: timing, waiting, computation or whatever) continues its operations so long as (while) some particular condition obtains; that is, the test for the exit condition is made *before* each execution of the body of the loop. Hence, a WHILE-loop can be executed as many times as needed, or not at all. An UNTIL-loop, on the other hand, has its test for the exit condition *after* the execution of the body of the loop and continues its operation up to the point where (until) the exit condition is reached. The fundamental difference brought about by placing the test after instead of before the body of the loop is that an UNTIL-loop, having no way of exiting from the loop until the test is reached, always executes at least once. When it is desirable to leave a loop unexecuted except in special circumstances, a WHILE-loop is preferable; but an UNTIL-loop is preferred wherever a guaranteed execution of the loop is desired. The needs of the individual program dictate which is the better choice in any given case.

All major compiler languages have specific commands for the purpose of establishing loops. Thus, one often hears BASIC programmers speak of FOR-loops and FORTRAN programmers refer to DO-loops. These commands simplify the establishment of loops for the compiler-language programmer but are of little use to the programmer working in machine or assembly language. For this reason, we are avoiding here any reference to specific compiler languages and their loop structures, since not all readers can be expected to have higher-order languages available to them on their private systems.

The Exit Decision

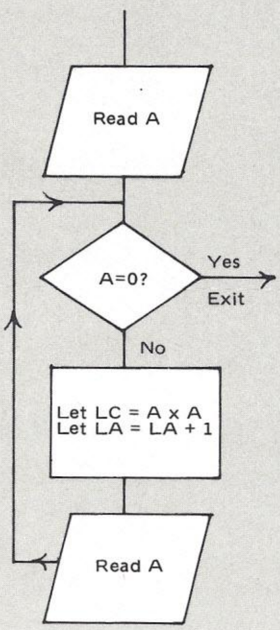
A loop is controlled by its *exit condition*. As loops are essentially circular in form, the exit is the only way a program in a loop can break out of the circle. Fig. 3a shows the basic structure of a loop, with the exit decision simply stated as Finished/Not-Finished. This decision can be set up in a number of ways. Fig. 3b shows the same loop where the exit condition is a specific case (A=10). In a loop of this sort, the Only Possible Exit is when the variable (A) has the value specified (10). An alternative to this kind of exit decision *uses a separate variable as a counter* recording the num-

B. Exit on Value of Data



This loop finds the squares of the first ten positive integers and stores them in ten adjacent locations.

D. Exit on Sentinel Value



This loop finds the squares of numbers given it as data and stores them in adjacent locations. It exits when the sentinel value of zero is reached.

Sample Data: 17, 4, 42, 71, 25, 0.

sentinel value 0.

ber of times the loop has been executed. Fig. 3c shows an example using this sort of exit. If a counter is used, it must be *initialized* at the beginning of the loop and incremented each time the loop has completed one full cycle of operation.

The use of counters to decide when to exit a loop has two drawbacks. First, if the value of the counter is altered by a programming error at other places in the loop (than the place at which it is supposed to be incremented or decremented), it will keep an inaccurate count and consequently fail to exit at the appropriate time. Second, and far more annoying, the programmer who doesn't wish to be restricted to processing same-sized batches of data each time he uses the program must alter the exit value of the counter every time a different-sized collection of data must be processed. Consider the program flow-charted in Fig 3c. Since the exit condition is C=10, only batches of 10 numbers may be squared during a program run. If the programmer wishes to find the squares of four, instead of 10, numbers he must change the exit condition to C=4; and if he wants the squares of 19 numbers, he must either change the exit condition to C=19 or process the numbers in two batches, adding one dummy number to the second batch so it would be 10 numbers long. This sort of continual modification gets very tiresome; most programmers avoid the need for it by only using counter-controlled loops in the sorts of cases where this problem doesn't arise.

One of the more convenient tricks for avoiding these difficulties is the use of a Sentinel Value. A Sentinel Value is a value selected to indicate the end of the data to be processed. Clearly, it would be pointless to select the number five as the sentinel value if there was any possibility of having that number appear as a piece of data; but if the data are, for example, all positive numbers, any negative number entered at the end of the data would serve quite well as a sentinel value. When using a sentinel value, it is simplest to place the test for the exit condition at the beginning of the loop. This permits the program to determine whether the last piece of data (the sentinel value) has been reached before starting its processing, thereby preventing any attempt to perform its operation on the dummy number which is the sentinel value. When the sentinel value is reached (for example, A=0 in Fig. 3d), it immediately exits to the next part of the program.

Any use of specific cases as exit conditions, however, has a major disadvantage; if anything prevents that specific value from being reached, a *continuous loop* can result, from which the program has no way of exiting. This can be embarrassing to say the least. One common safeguard against this problem is to test on *ranges* instead of *cases.* Thus, in Fig. 3c, one could rewrite the decision block to read C>10? This would cause any value of the counter which was greater than the desired number of cycles through the loop to result in exiting from it. Similarly, in Fig 3d, the test on the Sentinel Value could be rewritten so that when A⩽0, it would exit from the loop. One programmer set the exit value of his counter variable at zero and approached it, starting at an odd number, by decrements of 2. Naturally, his variable's value changed from 5 to 3 to 1 to -1 and so on, without ever having the exit value of zero. If he'd used the *range* of values C⩽0, he wouldn't have had this problem since exit would have occurred as soon as the value of the variable dropped below zero.

Mischievous Loops

Endless or *inadvertent loops,* caused by carelessness (as in the case above) or by accident or confusion, are one of the most common and troublesome bugs a program can have. They are easy to create, often hard to find, and lurk behind every subtle programming trick, waiting to catch the unwary programmer. Use of modular program design greatly reduces the risk of developing an inadvertent loop, but only extreme care can prevent their occurrence entirely. Higher level languages, especially those which, like Fortran and Basic, have a statement-number structure, are particularly prone to this problem. Yet, although every machine has unique hardware idiosyncrasies which can make the problem more or less severe, there are several common and avoidable causes of infinite and unintentional loops:

- *Testing for an exit condition which is either impossible* (like our fellow's C=0 condition), *or Which is at a constant non-exit value* (to increment the counter, by forgetting so

that it keeps its initial value throughout and never reaches the exit value).

- *Incorrectly entering a command so that the wrong address is utilized.*
- *Circular subroutine calls* (such as Routine A calls Routine B calls Routine C calls Routine A) or *Inadvertently circular pointers.* One programmer, on debugging an inadvertent loop in his program, found it to consist of 35 distinct pointers creating a giant circle! This is an extreme and subtle case of the A calls B calls C calls A problem.
- *A subroutine which calls itself,* with the consequence that, once entered, there is no escape.
- *Jumping to an address before that same jump command, when there is no intervention jump command* to prevent the initial jump command from being reached and executed endlessly.

Prevention Is Worth A Pound Of Print-out

The possibilities for infinite loops are limitless. There are, however, safeguards against them which can decrease the risk of forming one. On loops controlled by counters, initializing the counter immediately before entering the loop will eliminate the risk of its starting value being altered before the loop is reached. Similarly, making sure that the count isn't changed inside the loop except at the one place where it is supposed to be changed will keep the count accurate. Loops which delay program activity while waiting for something, such as keyboard input, can be saved from problems by the programmer's taking into account that something may prevent the input (or whatever) from happening. By specifying with a counter a maximum number of passes permitted through the loop before it drops through into the rest of the program, he can save a great deal of trouble. And, of course, all loops can be rendered easier to check and debug by the simple expedient of setting them off from the rest of the program with blank lines or indentations and by adding comments clarifying their functions. This alone can often help the programmer catch a flaw.

Loop Invariants

Another technique for minimizing the risk of problems with loops is the removal of *loop invariants.* A loop invariant is a mathematical expression inside a loop which is *not* changed by the loop's activities. For example, if the data the loop operates on is multiplied by X/3 at some point in the loop's operation, it is needless and wasteful of machine time to perform the division of the variable X (whose value does not change inside the loop) by 3 on every pass through the loop. It would be much more efficient to set the constant Y equal to the quotient of X/3 *before* entering the loop and multiply the data inside the loop by Y. Similarly, it is wasteful to perform the same calculation in several different ways within a loop; to test inside a loop for conditions which the loop's acitivity cannot change; or to include in a loop any computations or operations which the loop doesn't change and which don't affect the loop's activity. Making every loop as concise as possible will save machine time and reduce the chance for error. A thorough knowledge of algebra can also contribute to reducing machine cycles in calculation modules, since many complicated functions are defined as repeated applications of basic functions. Likewise, if it is necessary to use a loop that requires more than a page of code to write, it is wise to divide it into several shorter, more easily understood *modules* than to create a gigantic and potentially bug-ridden loop. The general rule is to keep loop structures as straightforward and coherent as

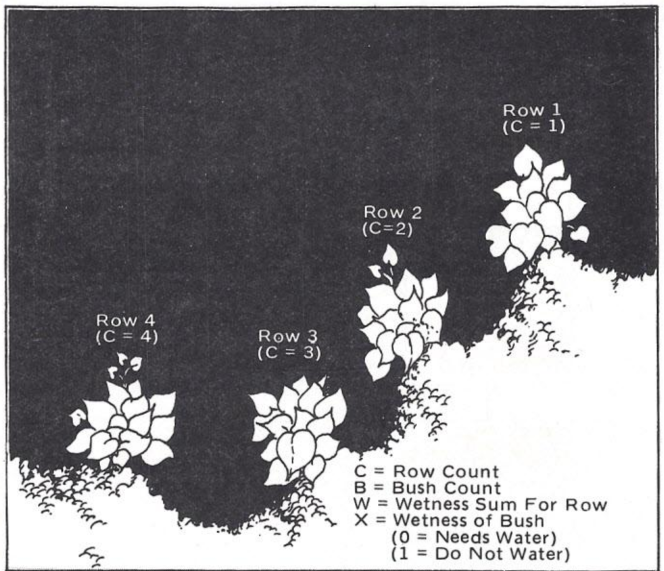
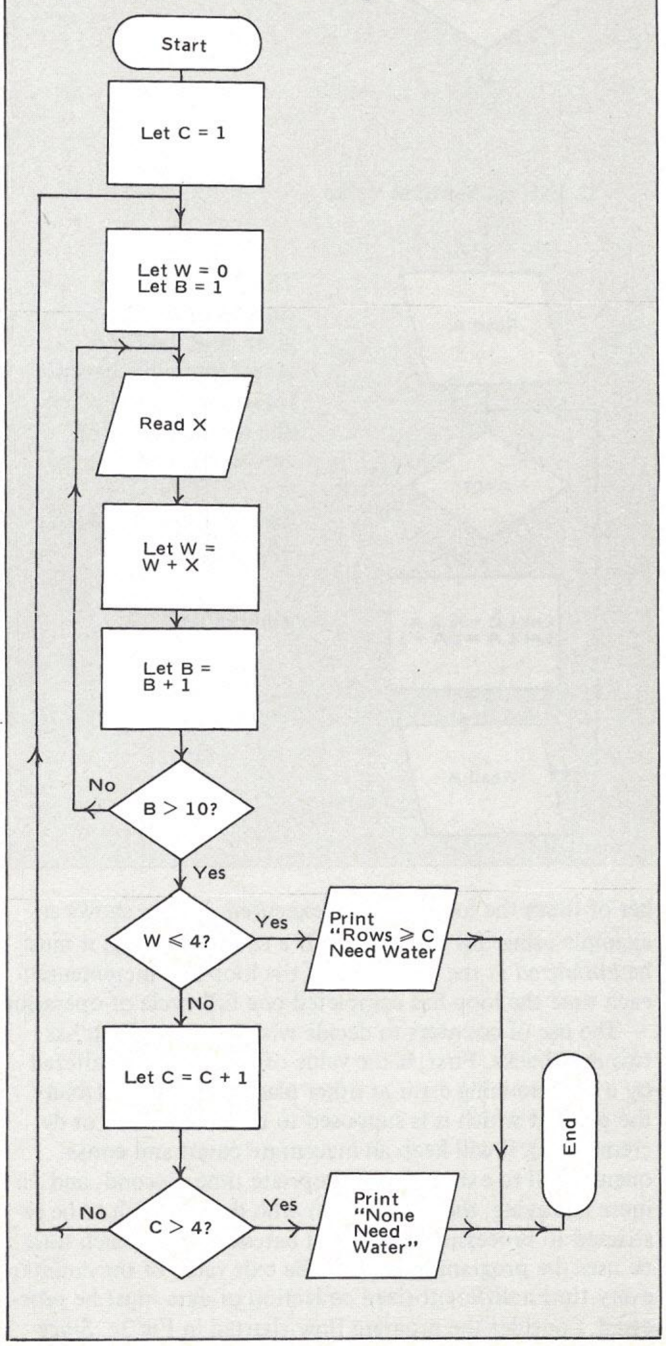


Fig 4 - A Simple Program Using Nested Loops



possible. Any program design which contributes to this goal will keep both errors and machine time to a minimum.

Nested Loops

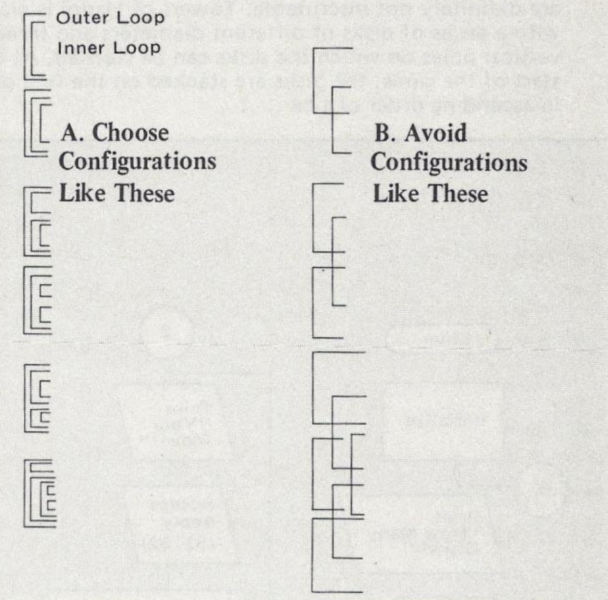
One of the more interesting uses of loops involves loops-within-loops, or, to use their technical name, *nested loops.* The purpose of nested loops is to handle more complicated operations than individual loops can perform. The innermost loop executes one or more times on each pass through the outer loop. Suppose, for example, that a programmer has four rows of rose bushes, each 10 bushes long, in his yard. He has a sensing device by which the moisture content of the soil around each bush can be independently measured and the results sent to his computer. To determine whether his roses need watering, he could use a program with a single loop which checked each bush in turn; but because he knows that the rows farther down on his sloping yard can have enough water, due to run-off, while the bushes farther up are dry, he uses a quicker solution. Since any row uphill of a row that needs water will also need water, it isn't necessary to check every bush in every row. Instead, he uses nested loops (Fig. 4). The outer loop has a counter which forces an exit from this loop after the last (farthest uphill) row has been checked (i.e., C>4). The innermost loop has a counter which terminates its activity when the last bush of any given row has been checked (i.e., B>10). In this design, the program will only check soil moisture levels until a dry row is found; the program cycles through the maximum (N) number of rows (C=4 in this example) if N-1 rows have been found to require no water.

As shown in Fig. 4, 0 represents insufficient water in the soil around a bush and 1 represents a wet soil condition. That is, the output of the soil moisture sensors is converted into a "water" or "don't water" binary form which is compatible with computer logic. The outside loop has this exit condition: if the variable W (Wetness), which is the sum of the wetness figures (0's and 1's) from the moisture measurement for each bush in a given row, is 4 or less, the program exits from the loop and outputs a message to the operator indicating that the row just tested (and all rows farther uphill) need water. Thus, without checking every bush (except in times of heavy rainfall, when all bushes would have enough water and, therefore, the "Needs Water" exit condition would not be triggered) the program determines when to water which rows.

Caution: Nesting Loops

Loops can be nested in many ways, a few of which are shown in Fig. 5. In some cases, the outermost loop will contain nothing besides the material required to set up the repeated operation of the inner loop, but sometimes it is desirable to include additional operations in the outer loop. In general, all rules, cautions, and tricks which apply to single loops apply equally to nested loops. It is worth noting, however, that the number of passes through an internal loop is the number of passes required by that loop (10, in the example just considered) *times* the number of passes required by the outer loop (up to 4 in the example); or, in the example, up to 40 executions of the inner loop. This may not sound like many operations, but if the programmer in our example mentioned above had had 40 rows of 100 roses each, running the program completely would require 4000 executions of the inner loop. While it is true that computers can operate at impressive speeds, it is worth bearing in mind that nested loops can so easily call for an astonishing number of executions that noticeable delays in response time to a single piece of entered data (up to half an hour in one program with which we are acquainted) can result. This sort of delay can sometimes (as in the example) be reduced by the introduction of an additional exit condition; but any use of multiple exits must be attended by the greatest of caution.

Fig 5 - Some Configurations For Nesting Loops



If improperly designed, multiple exits can lead to the uncomfortable situation of not knowing which exit caused the termination of loop execution on any given occasion. This can give rise to very frustrating debugging problems. Many of these difficulties evaporate, however, when the programmer inserts commands to output the key loop control variable values upon exiting. These print commands can, of course, be removed when they are no longer needed, but until then they are extremely useful.

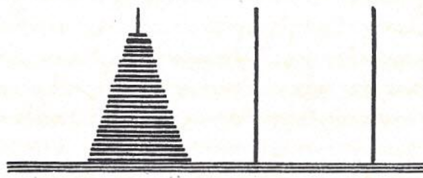
Loops Trade Time for Space

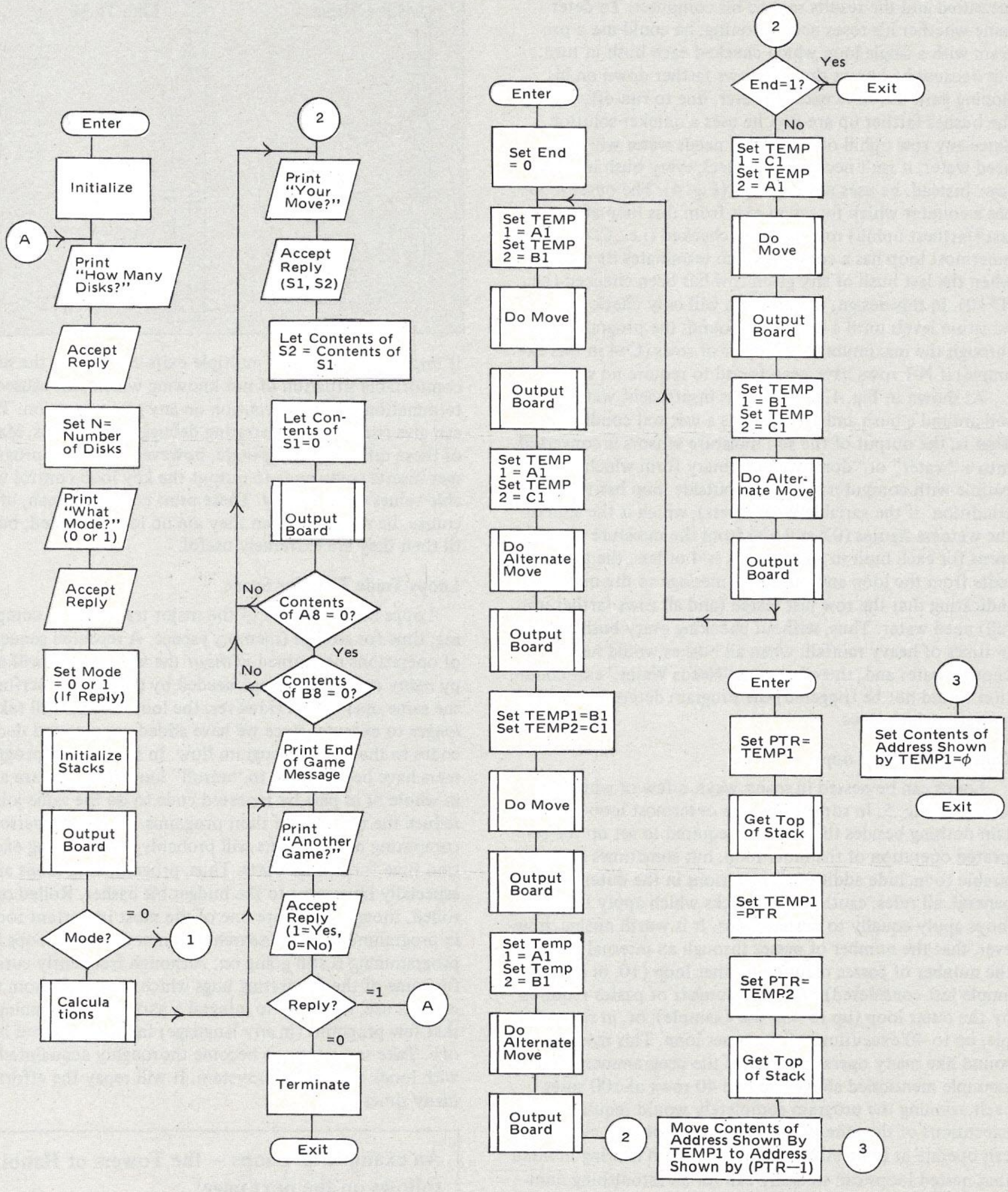
Loops introduce one of the major trade-offs in computing; time for storage (memory) space. A repeated sequence of operations performed *without* the use of loops will occupy *many times* the storage needed by a loop that performs the same operations. However, the loop version will take *longer* to execute, since we have added counter and decision codes to the normal program flow. In some cases, programmers have been known to "unroll" loops (i.e., replace a loop in whole or in part by repeated code to do the same job) to reduce the run time of their programs. Of course, personal computing programmers will probably favor trading execution time for storage costs. Thus, programming loops are especially important to the budget bit basher. Rolled or unrolled, though, loops are one of the most important tools in programming, and research into new forms of loops in programming is still going on. Although frequently cursed for some of the frustrating bugs which can result from their careless use, loops are so integral a part of programming that few programs (in *any* language) fail to employ at least one. Take some time to become thoroughly acquainted with loops on your own system. It will repay the effort many times over.

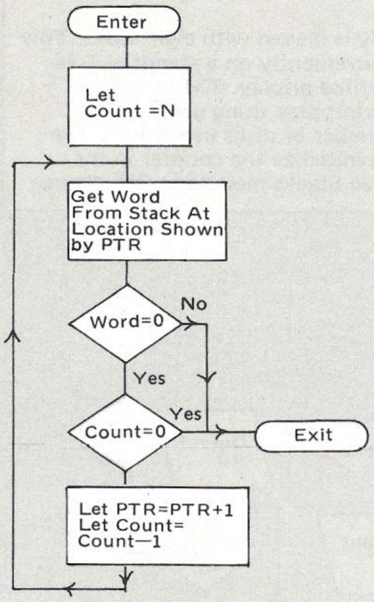
An example of Loops – the Towers of Hanoi – follows on the next page.

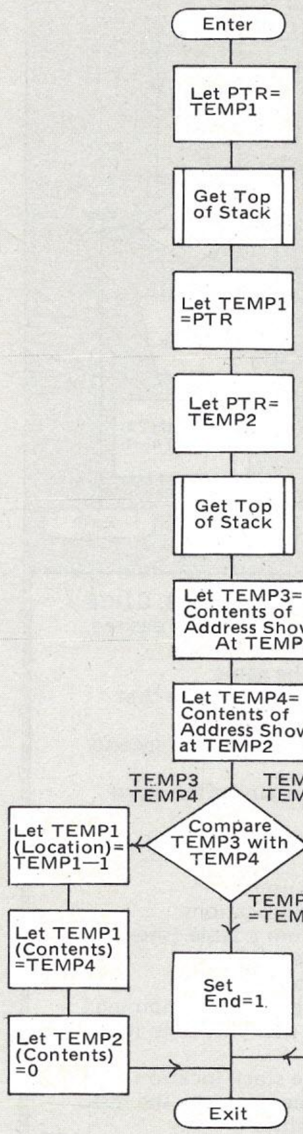
The Towers of Hanoi

The Towers of Hanoi, a popular puzzle, shows that, though loops may require careful implementation, they are definitely not inscrutable. Towers of Hanoi is played with a series of disks of different diameters and three vertical poles on which the disks can be stacked. At the start of the game, the disks are stacked on the first pole, in ascending order of size:

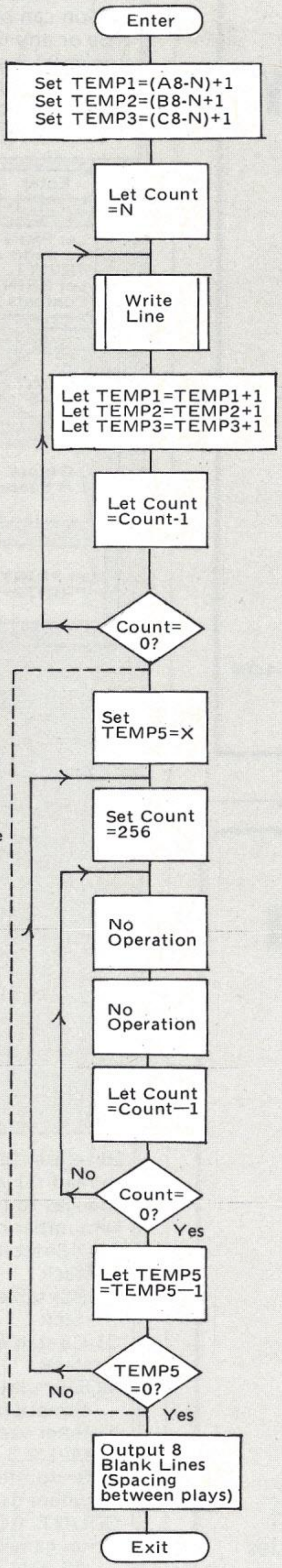


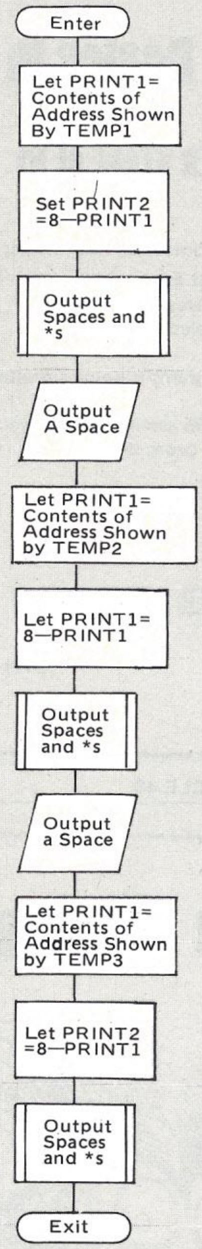


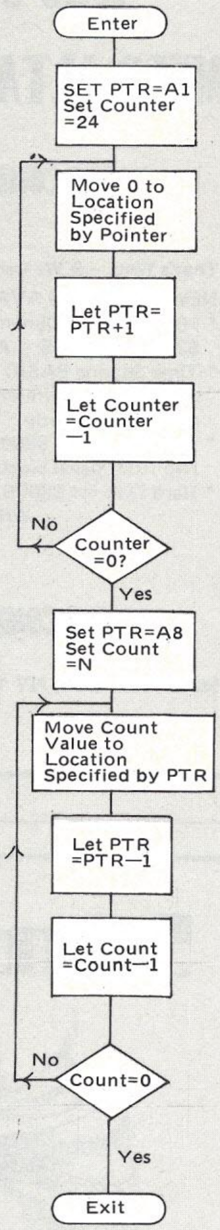




The section enclosed in a dotted line is a timing loop to provide a delay between printout out moves if a video display is used. Choose the value of X to give preferred delay time. Most assembly languages have NO OPERATION instructions, as shown: most compilers do not. If using a compiler language, insert a dummy addition (or whatever) to create a delay.





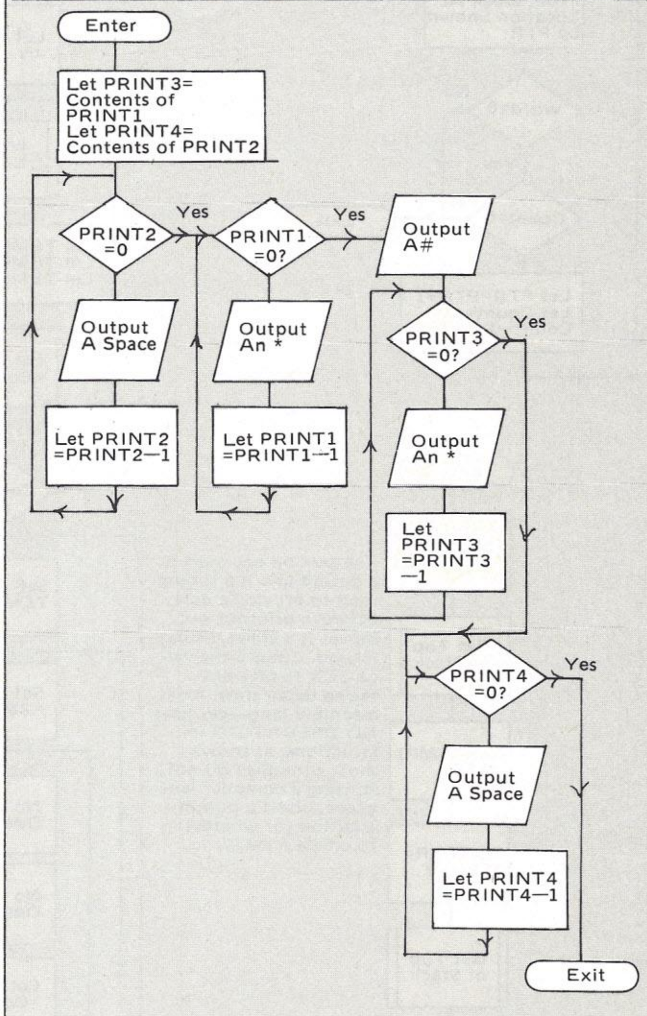


The object of the game is to transfer all the disks from the first stack (pole) to the last by moving one disk at a time. That sounds easy enough, right? Well, not quite — you cannot place a larger disk on a smaller one.

Towers of Hanoi may be completed in 2^N-1 moves, where N is the number of disks used. The algorithm for its solution is best understood by imagining that the poles form a triangle; on **odd-numbered** moves (1st, 3rd, etc.) the **smallest** disk progresses around the triangle,

one pole at a time and always in the same direction (e.g., clockwise); on even-numbered moves the only possible move remaining (i.e., **not** involving the smallest disk) is made.

The game traditionally is played with eight disks. This version can be played conveniently on a standard Teletype or any 64-character/line display. The following flowchart is set up to permit play using **up to** eight disks. To increase the number of disks used, set N=the desired number of disks; initialize the counter in the first block of the Initialize Stacks module to 3N; change



all references to stack locations A1-A8, B1-B8, C1-C8 to read A1-A_N, B1-B_N, and C1-C_N; and make desired changes to the output routines.

N=number of disks used to play the game
A1-A8=table of eight locations containing the first stack
B1-B8=table of eight locations containing the second stack
C1-C8=table of eight locations containing the third stack
MODE=player indicator (0=user plays manually, 1=computer plays the game)
X=timer variable (selected by the user)
TEMP1,2,3,4,5=temporary storage locations
PTR=pointer used to read data from a table (shows where data is temporarily stored)
COUNT, COUNTER=loop control variables
S1=in games played manually, the stack location (e.g., A5, B1) of the disk which the user wishes to move to another location.
S2=in games played manually, the stack location (e.g., C7, B2) to which the user wishes to move the disk.
PRINT1,2,3,4=print routine variables
END=end of game indicator (0=end, 1=game not over) ■

Deal Yourself in™...

Atlantic City, N.J.
August 27th-28th

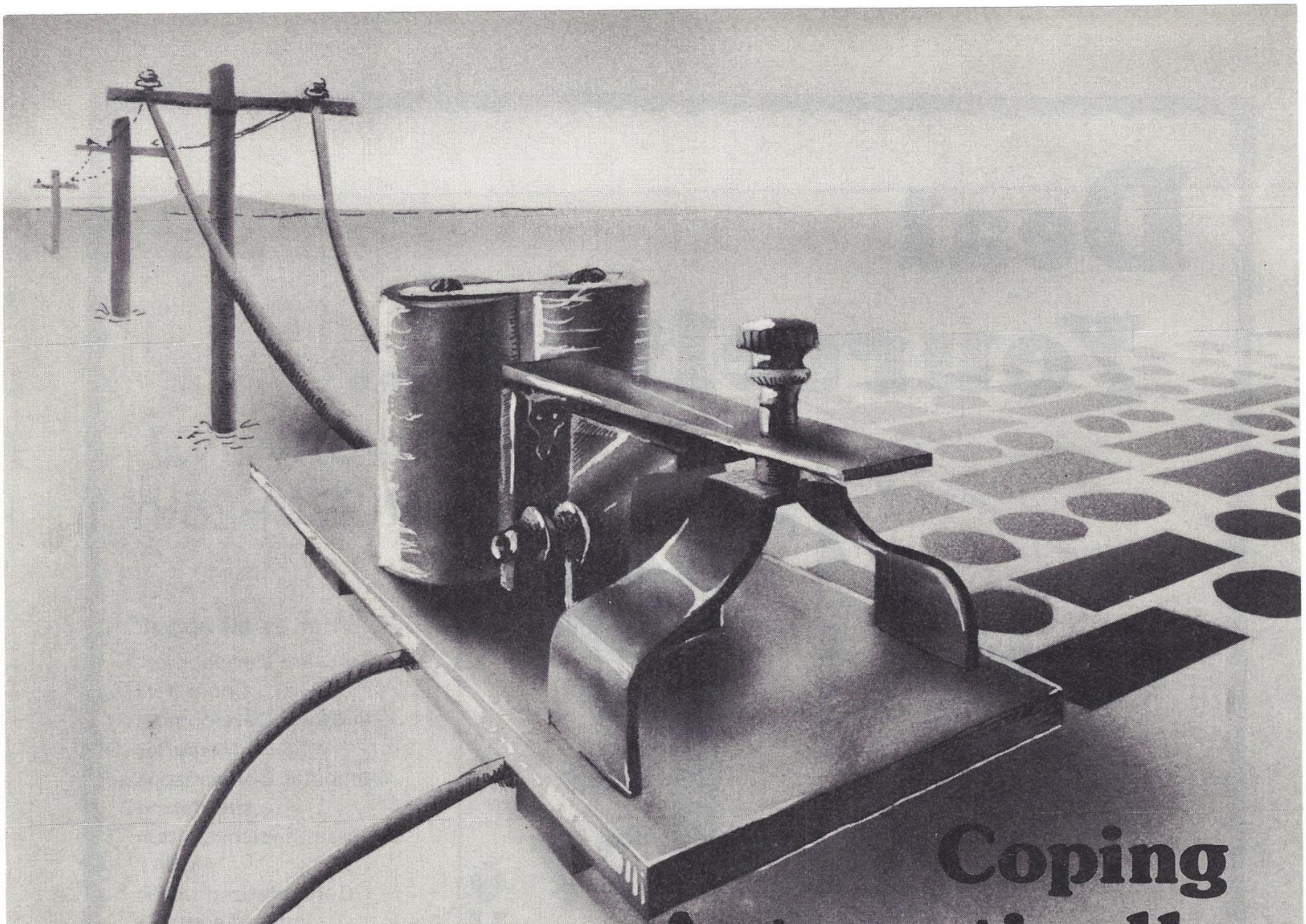
What its all about!

Software Development
Micro Computers
Hardware Development
Disc Memories
Computer Comparisons
Interfacing
Program Implementation
AMSAT
Computerized Music
Video Terminals
Kit Construction
Printers
Computer Games
Digital Tapes

- Seminars and Technical talks by leading electronic equipment manufacturers
- Major Exhibits from all over the country
- Demonstrations in many areas including Home and Personal Computing
- Door Prizes, Free Literature and Free Mementos
- All this plus Sun and Surf - Fun and Excitement - Relaxation and Leisure

Personal™
Computing
77 Consumer Trade Fair

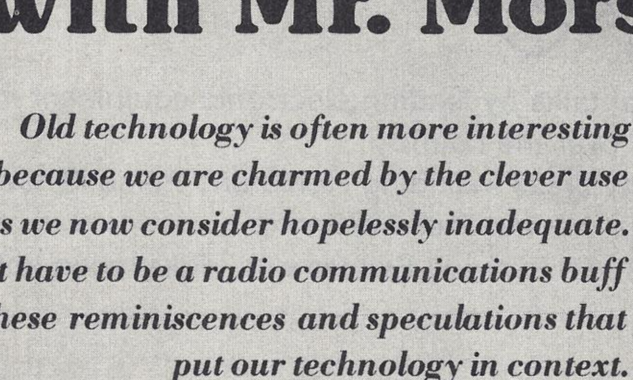
c



Coping Automatically with Mr. Morse's Code

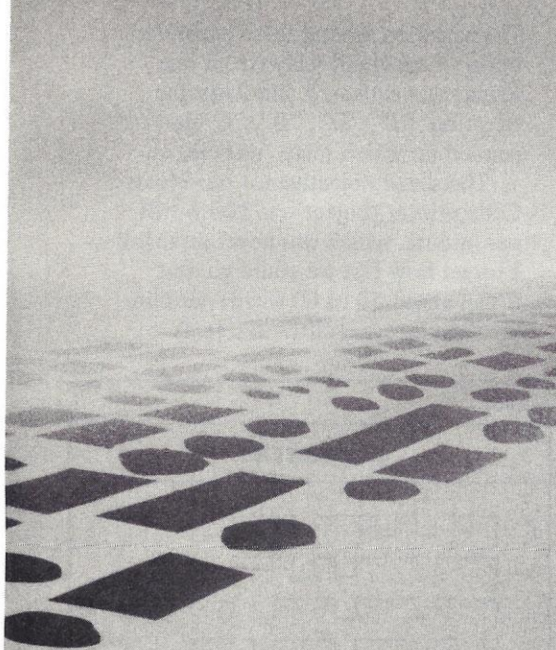
by Webb Simmons

Old technology is often more interesting than new, because we are charmed by the clever use of resources we now consider hopelessly inadequate. You don't have to be a radio communications buff to enjoy these reminiscences and speculations that put our technology in context.



Lately, there has been much activity in applying computers to handling the morse code. The most common use is to generate Morse Code characters from keyboard input to the computer or from a string of characters stored in the computer memory. Less often mentioned but of more interest to me is automatic recognition of Morse Code characters which are then converted to ASCII and either stored in memory or printed. I worked on this problem as an engineer decades ago, when vacuum tubes were still exciting.

Machines that generate Morse Code from a keyboard are quite old. Usually the keyboard device did not key the transmitter directly, but punched a two level tape that was later fed through a keying head that



keyed the transmitter. The tape could be punched very rapidly by a skilled typist, much faster than it was transmitted if the transmission was intended to be copied directly by a radioman. Some tape perforators were electromechanical but some were fully mechanical. The paper tape produced was similar to Fig 1

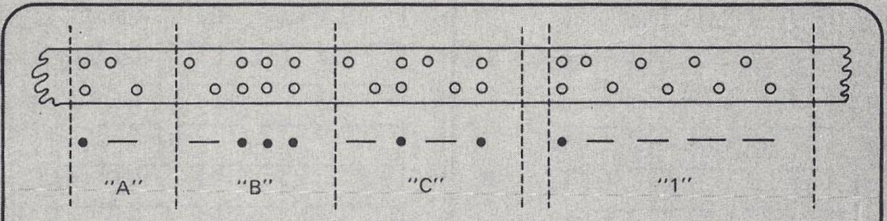


Figure 1 — Morse Code Perforated Tape

(the feed holes are not shown). Two holes in alignment produced a "dot" while two holes displaced by one hole spacing produced a "dash." The upper hole in each case starts an element (dot or dash) and the lower hole terminates an element.

The keying head had two "pecker pins" which examined the two levels alternately to start or stop an element as the tape was moved at a uniform speed through the head. The tape speed was variable over extremely wide limits from a slow speed of two or three words per minute to well over a hundred words per minute. Keying speeds to about 40 words per minute were intended to be copied directly by radiomen, but the higher speeds were recorded in some manner and transcribed later.

The technique just described was the state-of-the-art in 1940 and thereabouts. Earlier, there was a keyboard operated Morse Code sender that was fully mechanical except for the one contact closure that operated the transmitter. Pressing keys on the keyboard set up pins on the circumference of a circular cage much like a hamster exercise wheel. Depressing the keys advanced the cage wheel. Inside the cage was a mechanical "hamster" for the transmitter to be keyed continuously but, except for this, he could vary his speed without affecting the uniformity of the transmitted characters. It was the operator's average speed that was important. This was a most ingenious machine and is a collector's item today.

Transmitting Morse Code was fully mechanized prior to 1940 in the large news service agencies, in the large shore establishments of the U.S. Navy and elsewhere. The mechanization of Morse Code *reception* was something else and was crude in comparison.

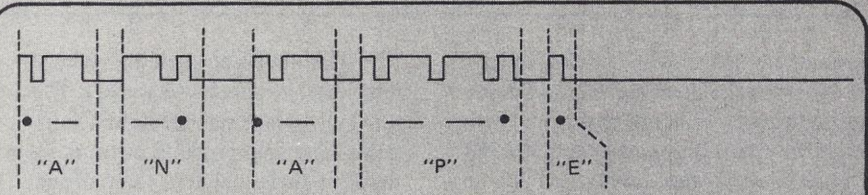


Figure 2 — Relative Timing for Morse Code

In the early 1940's, I had the fantastic good luck to join a project team whose objective was to interpret and print Morse Code without human intervention. The brains of the device was a combination of a few radio tubes and scads of relays. The output device was an early model IBM "electromatic" typewriter which had been modified so as to have two different keys for the letter "E." At the time, I was mystified by the need for the two "E" keys and had difficulty in understanding the circuit that drove the two keys. We called it the "double E" circuit.

To discuss the Morse Code printer, we must review the characteristics of the Morse Code. The basic time interval in Morse Code is the time the key is down during a dot (the "dit" of "dit-dah"). A dot is really two time units long because the interval after the key is down is filled by an equal interval with the key up. A dash (the "dah" of "dit-dah") is four time units in all with the key closed for three time units and open for one time unit. The space between letters is three time units and the space between words is five time units or greater (no maximum). These time units are shown in Fig 2 for the two words "an ape."

If we adopt the convention that every dot (on for one time unit) and every dash (on for three time units) is followed by an off interval of one time unit that is to be considered a part of the previous dot or dash, then the space between letters is two (extra) time units and the space between words is four (extra) (or more) time units. The shortest letter in the Morse Code is the letter "E" which is a single dot (one time unit) followed by its unit space for a total of two basic time units. The longest character is the numeral zero composed of five dashes for a total of twenty basic time units. The character zero is ten times longer than the letter "E." The "E" is half the time of the next longer letters and these are the letter "I" (two dots) and the letter "T" (a single dash). The "I" and "T" are four basic unit times.

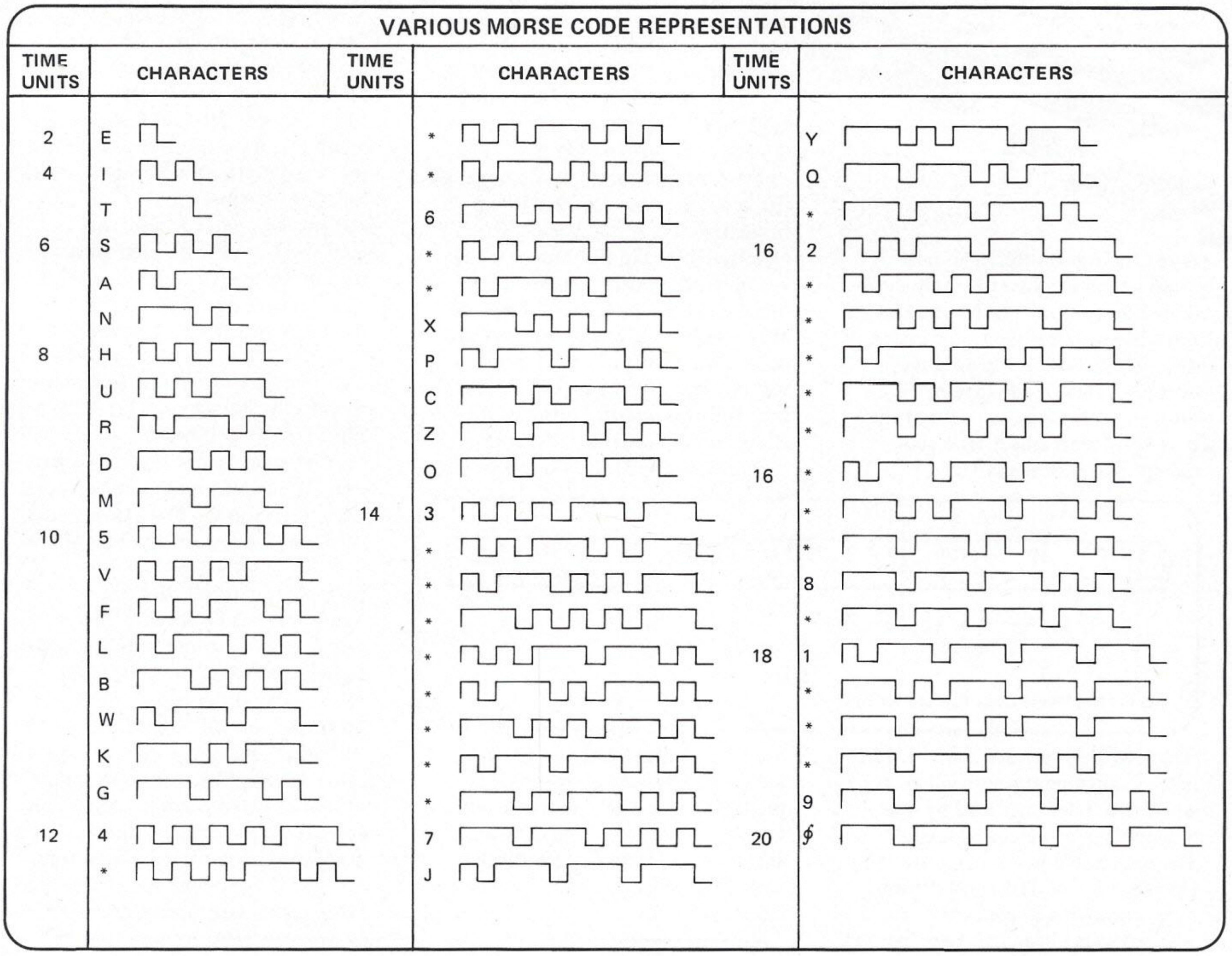
Our circuits assumed that all Morse characters are composed of some combination of from one to

five dots and/or dashes. Any combination of more than five was considered to be in error. Several combinations of five or fewer, such as dit-dit-dah-dah were also errors. Errors caused a special key to be struck. The logical inputs to our "brain" were from an element detector which responded to each dot or dash, a dash detector that did not see the dots (they were too short to qualify), an inter-letter space detector and an inter-word space detector. Upon an inter-letter space, a selected character (except the space bar) was typed and the "brain" relays were reset.

We had six relays in a chain that were so connected that if none were activated initially and an element (dot or dash) were started, relay number one in the chain would activate. The next element, dot or dash, would cause the number two relay to activate and relay number one to open, and so forth. Needless to say, relay number six caused the error indicator to print (but not immediately; printing was upon the inter-letter space). The character selector circuit was called a "christmas tree" circuit. At the head or top (in the first element position) was a single-pole, double-throw relay. This relay was activated (and held activated) if a dash was received while chain relay one was activated. The next christmas tree relay had two poles (all christmas tree relays were double throw) and was activated if the second letter element was a dash. The third christmas tree relay had four poles. The two possible outputs from the first christmas tree relay drove the two poles of the second christmas tree relay, etc., until finally the fifth relay had 32 possible outputs. To put it in modern terms, it was a 5-bit decoder – or was it?

Actually, it was a variable-number-of-bits decoder. If, for the moment, we consider dots as zeroes and dashes as ones, all zeroes could mean any of the characters, "E", "I", "S", "H", or the numeral "5." The selection among these depended upon which chain relay (from one to six) was activated. Similarly the selection "N", "D", "B", "6" depended on which relay was closed.

The speed objective for our Morse Code printer project was 100 words per minute, which was never attained. I forget how fast we could go, but it was about 50 or 60 words per minute, maybe even a little faster. It was many years ago. For those of us who were former professional radiomen, it was impressively fast. The thing sounded like it would fly apart with all the clickity-clack from the relays added to the clickity-clack of the typewriter.



The table above shows legal characters for various Morse Code combinations from the shortest character ("E" in two time units) to the longest (zero in 20 time units). Combinations with more than five elements are not shown although many have commonly accepted meanings such as "end of transmission" (dit-

dit-dit-dah-dit-dah). Some combinations caused an asterisk (meaning "error") to be printed even though the combination was meaningful, such as the "slash" (dah-dit-dit-dah-dit), and there was a key for it on the typewriter. No one is perfect.

The table below suggests the average length of Morse Code characters is 8 or 10 time units. The shorter codes were selected, generally, for the most-used characters although there are some strange selections. The letter "O" is a frequently used letter but is 12 units in length, thus longer than "V" or the numeral "5." When Morse Code is transmitted at 50 words per minute on the average, the instantaneous speed for the letter "E", when it occurs, is 200 words per minute or more. This high speed for the letter "E" is made even more troublesome because many words contain a double "E." Examples are "free", "speed" and so forth. To print consecutive "E's"

TIME UNITS	NUMBER OF CHARACTERS
2	1
4	2
6	3
8	5
10	8
12	7
14	5
16	2
18	2
20	1

at high speed, we had those two keys with the letter "E" physically far apart in the type basket of the typewriter. The keys were used alternately for the "E" regardless of how they showed up in the text. The "E" select line from the christmas tree circuit was directed to the relay equivalent of a flip-flop circuit which we called the "double E" circuit. As I recall it, this imitation flip-flop used four relays. What we really needed was a buffer in the core memory of a computer but, alas, computers had not been invented yet.

I will now review the troubles we had for the benefit of those who wish to experiment with the problem of a computer reading Morse Code. The exercise may be entertaining, though Morse Code is not often used these days. First of all, it was necessary to get rid of noise in our signal caused by static, fading signals and so forth. After converting the signal to a D.C. signal by demodulating a keyed audio tone, we used RC circuits in various ways to eliminate short "on" voltage spikes during an "off" state and to fill in short "off" dropouts during an "on" state. After cleaning the signal, we squared it up with a limiter circuit and fed it to our "detectors." The detector which responded to "elements" (dots and dashes) was simple, straightforward and trouble free with no runtime adjustments needed. We had three adjustable detectors that were set with front panel potentiometers. These were the dash detector, the space-between-letters detector and the space-between-words detector. Ideally, these could have been ganged to operate with one control, but it did not work out that way for us. These detectors were adjusted to a particular transmitting speed.

For machine-sent Morse Code that was not more than about 30 words per minute and with a good clear signal, we could copy all day with few errors. However, it was devilishly difficult to copy hand keyed Morse Code. Those of us who claimed to have a "good fist" were shamed when we attempted to send to our Morse Code printer, even when we sent slowly. We quickly learned that there were different "optimum" adjustments among the detectors for each of us when sending by hand. Unfortunately, our machine did not correct itself for varying speeds and, in fact, made no automatic adjustments of any kind.

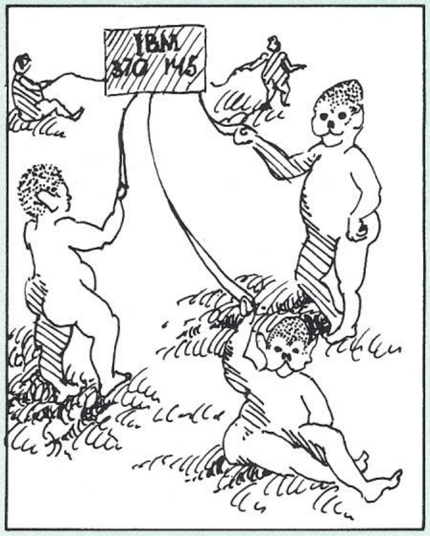
To perform this job with a computer, the program should measure the length of every "on" time and every "off" and continuously readjust its criteria for the three critical time lengths. A particular person's timings will be peculiar to himself and will vary both relatively and absolutely depending upon his speed, tiredness, mood or whatever. A guess on my part is that when a new signal is tuned in and directed to the computer, the peformance at first will be erratic or even pure nonsense, but the computer's performance will rapidly improve as it "learns" this person's style and will then follow his performance as he speeds up and slows down if the change of pace is not too rapid. [In fact, some elegant systems now perform just this way. Ed.]

Where is that time machine? I must whip back to the 40's with a pocket full of 8080's.

Just A$k

More than a thousand customers each day personally tap into the big IBM 370-145 at the San Diego Federal Savings and Loan Association to examine the status of their own accounts and loans. The system is called A$K.

Terminals in 22 offices of the company allow customers to insert an ID card, punch in a four-digit coded personal "password" and ask for information. He can key in questions about his current savings balance, the last transaction date and amount and interest earned in the current quarter,



the current year and the previous year. In addition, the read-out display can also tell the customer his current home loan balance, interest paid this quarter, in the current and previous year, and the amount and date of his next loan payment. The system saves time for both tellers and customers, providing more information, in convenient form than the customer ordinarily requests.

Obviously, steps have been taken to prevent the customer from altering records in any way. All he can do is look at them. The way the system is set up, it's very difficult for anyone **but** the customer to see the readout and learn about his private business.

If you have ever stepped into a pay phone booth different from Ma Bell's standard system and tried to make a call (especially one of those phones into which you put money **after** dialing the call and getting the connection; coins entered in advance of the connection don't count and the other party, saying Hello, Hello? Hello! can't hear you until you do the right thing) you know how great is the technical challenge of dealing with unfamiliar technology.

IBM's Human Factors Laboratory was given the task of developing a simple, step-by-step procedure for customer operation of the terminal. So far, so good. Perhaps a significant part of our population will learn to be at ease with personal computer operation by playing with the system down where they keep the money.

Quiet!

A new Power Line Filter has been announced by Electronic Specialists, Inc., designed for use where microprocessors, teletypewriters, TV games or other interference enters the power line. The equipment is also useful where TV, FM or other listener equipment picks up interference from the AC lines. $10.50 buys a 350 watt, 2-conductor unit; $13.50 buys a 3-conductor unit. The company is at Box 122, Natick, Mass. 01760.

Does this make you glad? Do you have interference problems? Would you even know it if you did? If you are an average non-technician, you probably have only a very foggy notion of what this is about, but RFI (Radio Frequency Interference) or EMI (Electromagnetic Interference) is a very important matter in our modern world. Our computer systems, television sets, automobiles and most other implements we use are tightly bound up in complex regulations controlling their RFI output. The fact is that almost every activity, especially any involving electricity, puts out broadcast or line transmitted noise that bothers other systems. Even if you fling a rock down to the ground, the stressing of its physical structure causes output of an electromagnetic pulse that may be detected in some electronic system nearby.

Remember the WWII movies in which the fighter plane squadron commander orders "radio silence" and the guys all began glancing at each other out of their cockpit windows? That order didn't just mean to quit broadcasting; it meant quit **receiving**, too. Every radio actually rebroadcasts the signals it receives at some small level. The flyers worried that those rebroadcast signals could be detected and the location of the squadron determined by the enemy.

Computers put out such RFI, too. An old gag observes that if

random access

the guidance system for Washington's National Airport were to fail, planes could just as easily steer themselves in by homing on the broadcast signal output of the computer installation at the National Security Agency. Notice that a spy doesn't have to wire himself into your computer to figure out what you're doing with it. Information is where you find it.

If you switch on your calculator next to your portable radio, you may induce a great noise in the radio. If you program your computer according to available schemes, you can induce "music" in a radio set next to the computer, within a foot or two, controlling the tones of the interference. The FCC is satisfied to keep this interference down to a dull roar, given the impossibility of complete suppression. That's what this power line filter is all about; you don't want to pump a lot of noise into the power lines.

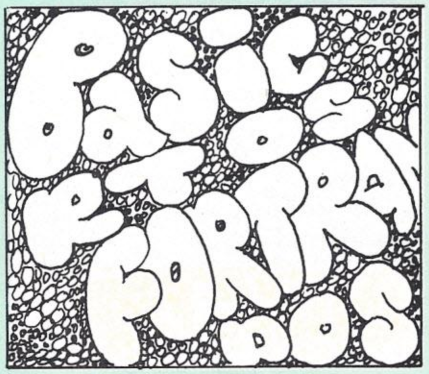
Perhaps the cleverest use of this low-level RFI phenomenon is provided to us by Processor Technology. When you get your SOL, it comes with a Star Trek tape. That program not only keeps track of the Enterprise's activities and controls complex images on the video monitor, but controls incidental broadcast signals from the computer.

Thus, when you fire photon torpedos or lasers, or when an emergency develops and alarms are set off on the spaceship, the player can hear appropriate sound effects on a radio placed next to the computer. That's almost like something for nothing. Very clever.

Do you need that power line filter? Only if you are bothering somebody or being bothered yourself. Most of your equipment is already equipped with mechanisms for keeping the roar dull.

Toe in the water

Until now only IBM, among the big, established computer manufacturers, has made any significant gesture toward the personal computing market (with its 5100 system). Now one of the major manufacturers of minicomputers is



taking a whirl at selling through retail computer outlets.

Data General Corporation has made an agreement with THE COMPUTER STORE, a chain of retail personal computer outlets in the northeastern U.S., to supply its entire Micro-NOVA line of microcomputers, peripherals and software. Beginning immediately, The Computer Stores will offer Micro-NOVA expansion memories and interfaces, DOS, RTOS, BASIC, FORTRAN and development software, the new DASHER matrix printers, as well as the 6000 line of video display terminals. All on-site field service and maintenance contracts available to Data General's customers will be offered through THE COMPUTER STORE.

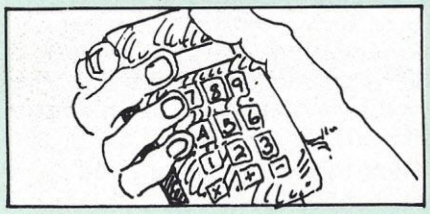
This is not bargain basement equipment, but straight professional gear designed for commercial application. Prices are not small, but realistic. The novelty lies in offering the systems to a different sort of market in the general public. Relations between minicomputer manufacturers and their commercial customers are often abrasive and maddening. This experiment should prove very interesting.

Teacher's pet

CALCULATORS/COMPUTERS is a new magazine designed to fill what its publishers regard as a distinct void in the availability of practical computing materials written for educational purposes, considering the booming distribution of hand-held calculators and the present surge in computers for the home.

Each issue of the magazine is to include "concrete, self-contained instructional units which can be directly copied and used for instructional purposes. Material will range from elementary school level through community college. A teacher's commentary or guide will accompany each unit. Some units will be for hand-held calculators, others for BASIC computing."

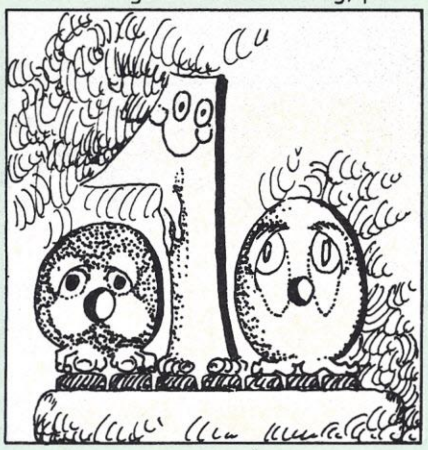
The real novelty in this is that the publisher not only fails to forbid copy and use of the material, but **encourages** teachers to copy out worksheets and other materials for classroom use, laying out the material for the purpose. COMPUTERS/CALCULATORS is designed to be an immediately useful educational tool.



Both octal and hex

Texas Instruments is test marketing a new calculator designed specifically for computer programmers. The $49.95 device works in three different number bases, including hexadecimal (base 16) and octal (base 8), as well as decimal for everyday arithmetic.

Since most computers use a binary number system for internal data storage and addressing, pro-



grammers and computer users are often required to convert and manipulate coded numbers in other related number bases. In the past, this required time-consuming manual calculation and the use of addition, multiplication and conversion tables.

Now users of the new TI calculator can get those answers fast by keying in the problems in the same number system used by the computer. Then the calculator performs its operations in that number system and converts results to a number system which users can interpret more rapidly.

For example, a computer may store a 32 bit code containing that many "ones" or "zeros" in binary form to represent either data or internal instructions. In the hexadecimal number base, that same code shrinks to only eight digits which can be operated upon and stored in the calculator memory just as these same things are done with the longer code in the computer system.

Among other typical applications, the TI Programmer will convert memory addresses to decimal form, add relative addresses to a base address to find specific computer memory locations or determine if there is enough space in the computer's memory to hold a new block of data. Also, the arithmetic and logical functions of the calculator can emulate internal computer operations.

The calculator's parentheses keys help evaluate complex expressions without requiring the user to store or write down intermediate results. The parentheses specify the order of execution of operations in a problem. Up to four operations can be pending at any one time and these can be a mixture of arithmetic or logical operations in any combination of the three number bases.

The calculator can perform bit by bit logical operations on numbers in hexadecimal or octal. Included are AND, OR, Exclusive OR and SHIFT operations.

Users will also find the calculator useful as a regular day-to-day, four-function machine in the decimal base. Its speed in all number bases is essentially instantaneous.

For information, write to Texas Instruments Incorporated, Inquiry Answering Service, P.O. Box 5012 M/S 84 (Attn TIP), Dallas, Texas.

Displaying the wares

More retailers are taking time from the daily hysterical press of business to prepare sales literature that is genuinely helpful to the naive prospective customer. This breaks a well-established tradition of handing the prospect a list of technical specifications for a system that is totally incomprehensible to him.

The BYTE SHOPS OF ARIZONA (two stores around Phoenix, another in Tucson) have now issued the Byte Shopper, a forty-page catalog/book to warm the heart of the browser who has no idea what he might want to buy.

The 11" by 14" pages of the catalog are full of big, helpful pictures; a modest glossary is included; and numerous alternative full systems are described in terms that give the novice a good idea of what he can buy for how much, then what he can do with what he has bought.

The Byte Shopper is not produced with National Geographic

quality, but it's a great early effort to help the customer.

The hooker — they don't give it away. If you want the Byte Shopper, you'll have to send two dollars to P.O. Box 28108, Tempe, Arizona 85282.

random access

Backup

Computer Kits, Inc., a retailer in Berkeley, California, has taken the plunge and is offering a one-year parts and labor guarantee on all assembled systems and components sold by the store. More, the store offers a four-year service contract beyond that at a cost of $60 to $90 a year. "This is the same as a service contract one might purchase from a major business machine company," comments Pete Roberts, president of Computer Kits.

"From the beginning, it was our intention to stock and sell only commercial grade parts and components. To do this, we made purchases from numerous manufacturers, tested for quality and reliability, and in the end, rejected about 90 percent of what we tested.

"The end result is that we sell only the best, which means we can offer our customers a meaningful guarantee without fear of going broke."

For customers who live beyond comfortable driving range of the store, of whom there are many, most still bundle up their hardware and ship it in for repair, enduring a deal of nuisance and expense. But this guarantee is a notable experiment for which all personal computing enthusiasts may hope the best.

As one technician said, "If we can get a system up and working at all, it will probably keep on working for years. Certainly, if it gets past the first ten hours, the chances are very good."

Computer Kits great secret is assembly of the original systems in its own shop, where people have learned to solder neatly, and plug things in properly. Chances are that the people who built the system will also work on its repair. That personal relationship is important.

High figures

Bob Benedetti, an investigative reporter for CFCF television in Montreal, Quebec, and former helicopter-based traffic reporter, has designed a series of programs for his HP-65 programmable pocket calculator that can provide station listeners with extremely accurate forecasts of the time they will need to reach their destination. Using the program, Benedetti can even compare different routes and suggest the most time-saving combination.

Tutorial Basic

In this day of Computer Aided Instruction, when computers are opening new worlds of learning to interested students all over the Western World, you may have wondered why nobody was using Computer Aided Instruction to teach programming. Well, somebody is. If you have access to a 6800 system, you may be able to put yourself through an instructional procedure that takes the mystery out of BASIC.

Computer Software Services is offering software on cassette that transforms your computer into an instructor on 8K ANS BASIC. Each 5K lesson, accompanied by lesson plans, will coach and prompt the user through the BASIC commands and programming techniques. Part I presents the fundamental commands (PRINT, INPUT, LET, DATA, IF-THEN, FOR-NEXT). Parts II and III teach the concepts of formatting (using CHR$, LEN, etc. . .) as well as examples of reading and writing characters to cassette via BASIC and MIKBUG. The lessons run in a 12K

CSS offers a sampler package of products including cassette listings and full documentation of three programs: Lesson I of "Learn BASIC," "entertaining game" and "educational/program game," for $6.95.
Computer Software Services, 830 1st St., Encinitas, CA 92034.

cor~tex

by John and Mickey Kogut

With the phenomenal development and growth of the pocket calculator industry in recent years, it seems that new calculator accessories and peripherals are appearing in the market every day. If this trend continues, you might be seeing advertisements similar to this one in your favorite technical magazine.

USE YOUR HEAD – USE

COR-TEX

A SATISFIED **COR-TEX** USER

You probably know that most people use only about 10% of their total brain capacity. Why not put this undeveloped resource to work for you adding valuable memory registers to your pocket calculator?

Now with **COR-TEX**, Alamo Industries' revolutionary new neural adapter, you can do just that. Yes, with **COR-TEX** you can access up to 0.36 cubic centimeters of pure grey matter for use as calculator memory. And now the **COR-TEX** neural adapter incorporates an important new override feature; MINDBLOK. MINDBLOK is a cerebral safety interlock which protects **COR-TEX** users against accidental neural intrusion into the brain areas controlling personality development, motor skills, and involuntary muscle control.

With MINDBLOK, **COR-TEX** users are guaranteed thousands of hours of trouble free computation with a 0.997 probability of no permanent brain damage.

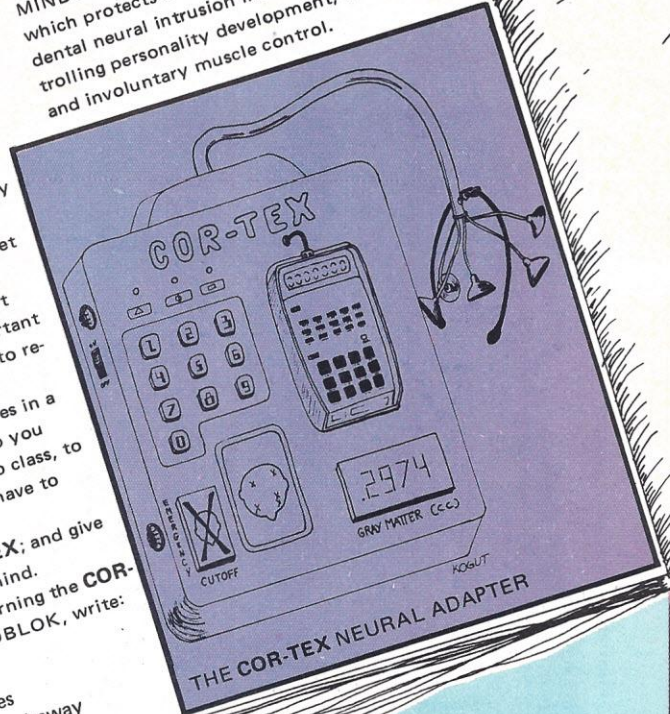
And **COR-TEX** is so easy to use. Just attach the cerebral contacts to the appropriate areas of your skull, set the MINDBLOK safety interlock, and you're ready to add up to 10,000 new memory registers to your pocket calculator. This added memory is great for those problems your calculator alone can't handle, or for just storing all those important telephone numbers you just can't seem to remember.

The **COR-TEX** neural adapter comes in a handsome leatherette carrying case so you can take your **COR-TEX** with you to class, to work, or anywhere else where you have to think on your feet.

So why don't you get **COR-TEX**; and give your calculator a piece of your mind.

For more information concerning the **COR-TEX** neural adapter with MINDBLOK, write:

COR-TEX
Alamo Industries
1650 Medulla Parkaway
Hoos Tune, TX 76133



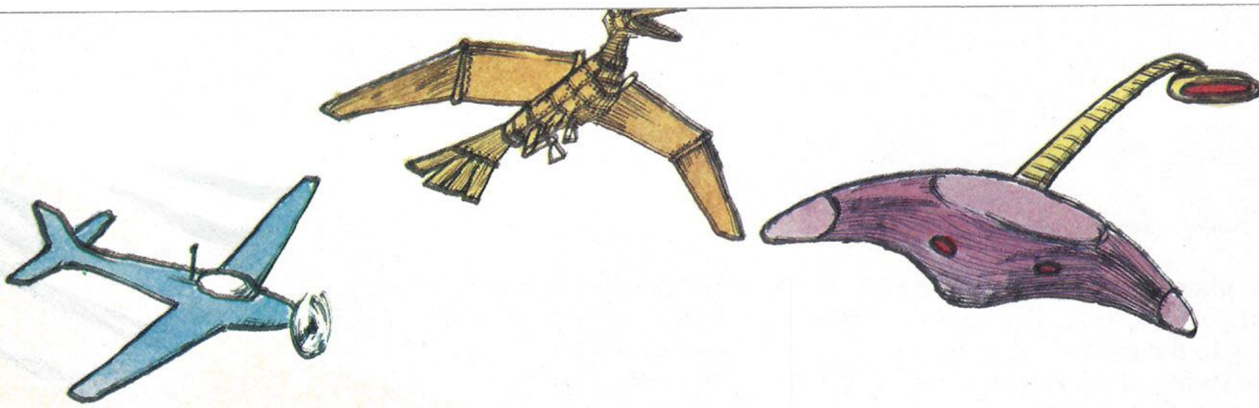
THE **COR-TEX** NEURAL ADAPTER

While the product in the ad is still probably imaginary, an analogous system is startlingly real. Artisan Electronics Corporation has recently announced its Microcalculator Model 85, which taps into the larger memory of a microcomputer much as the Kogut Cor-Tex device taps into the brain.

The Model 85 is provided as an assembled circuit board that can be interfaced with almost any 8-bit microprocessor. The computer, unlike the average human being, doesn't object to wearing the calculator all the time. Input to the calculator is through the computer-control keyboard. Output may be through a printer or video display. The Model 85 is a nice programmable calculator with many scientific functions.

The key phrase in the literature that makes the calculator relevant here is: "Most scientific calculators, which are programmable, permit only 100 steps per program. With the Microcalculator and microprocessor with cassette interface capabilities, programs are limited only by the RAM memory of the microprocessor system."

Artisan is at 5 Eastmans Rd., Parsippany, New Jersey 07054. Don't bother to ask them about Cor-Tex, but they'll be glad to tell you all about the Model 85. ■



WHAT IS A ROBOT?

As the author of BUILD YOUR OWN WORKING ROBOT, David Heiserman has been hard-pressed by many critics to explain in a single sentence what he means by "robot." Science fiction history notwithstanding, the whole subject is new and ill-defined. So, in about fifteen hundred words, the author makes a stab at definition, carefully sidestepping some issues that make him and many other people uneasy.

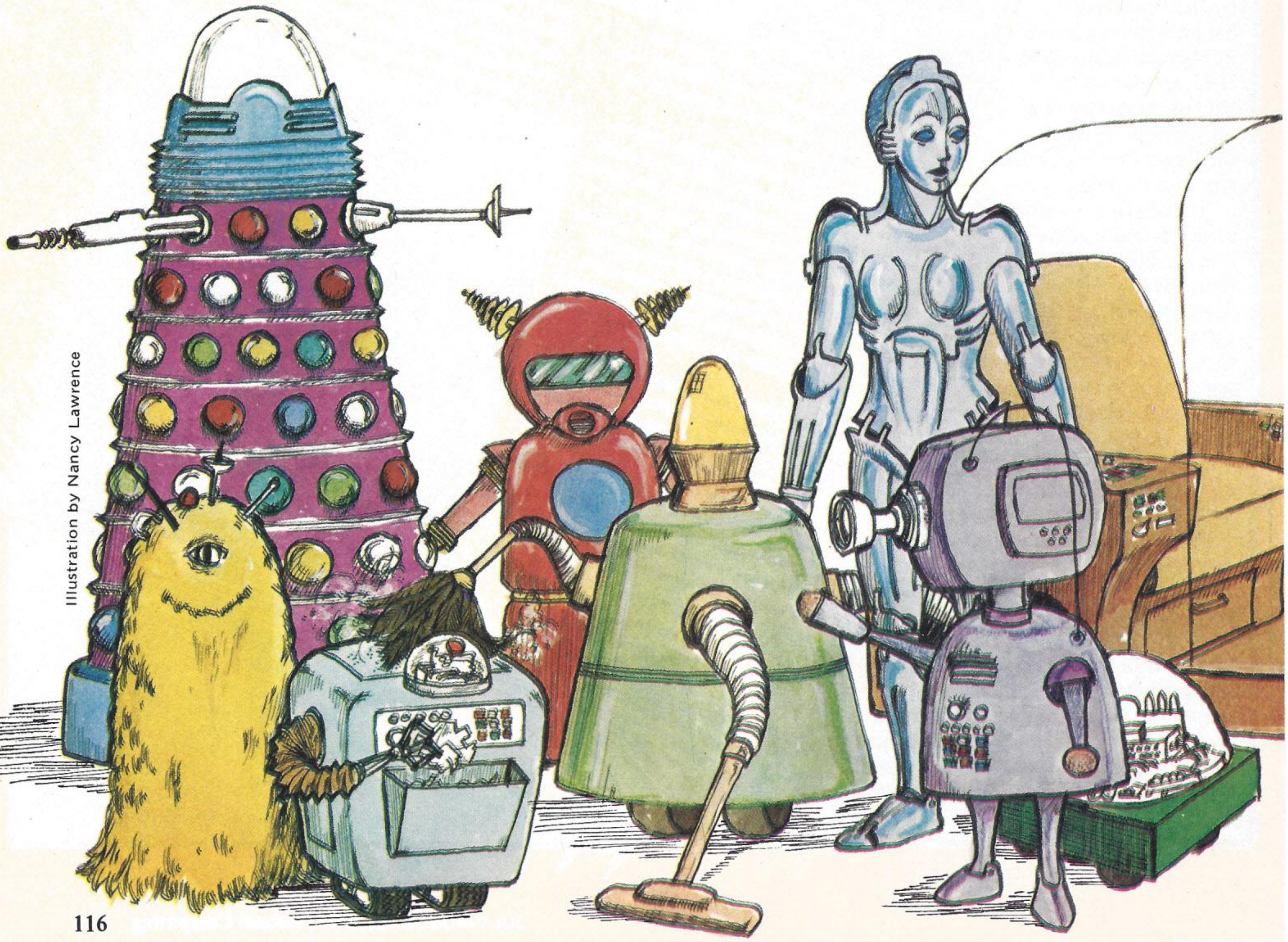
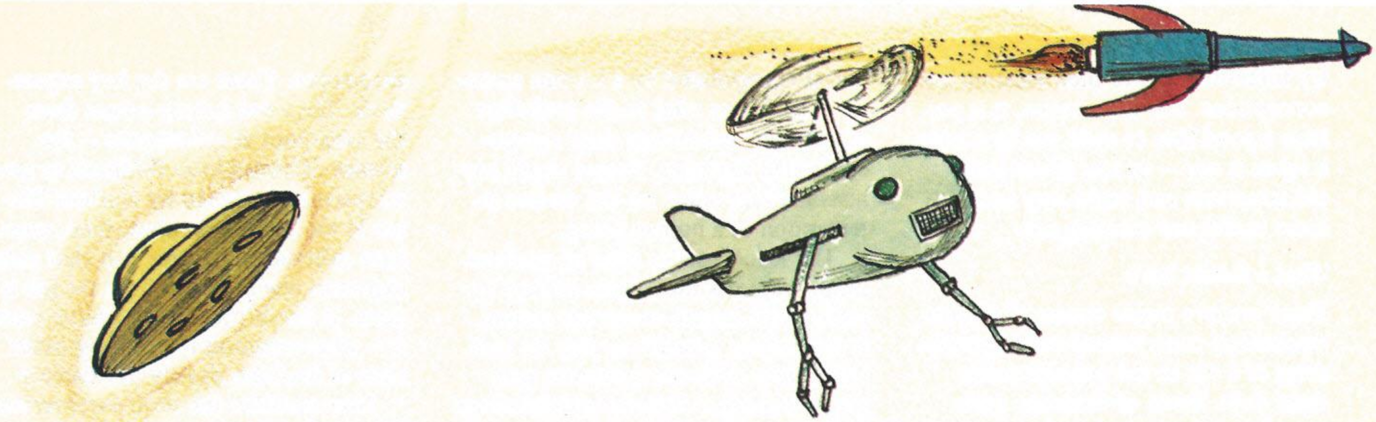


Illustration by Nancy Lawrence



By David L. Heiserman

There appears to be a groundswell of popular interest in robotics these days. Of course imaginary robots have been marching through the pages of science-fiction stories and across film and TV screens for quite a while now. But this new interest is not a matter of pure imagination. Rather, it has all the earmarks of a genuine technological revolution in the making.

The evidence for this new interest in robotics comes from several different sources. Consider the successful formation of a vigorous new organization, the U.S. Robotics Society (USRS). Then look at the growing number of robot-like machines appearing on popular TV talk shows and the evening news these days. Finally, consider the fact that a do-it-yourself book on the subject, *Build Your Own Working Robot* has gone through two printings in a little over one year.

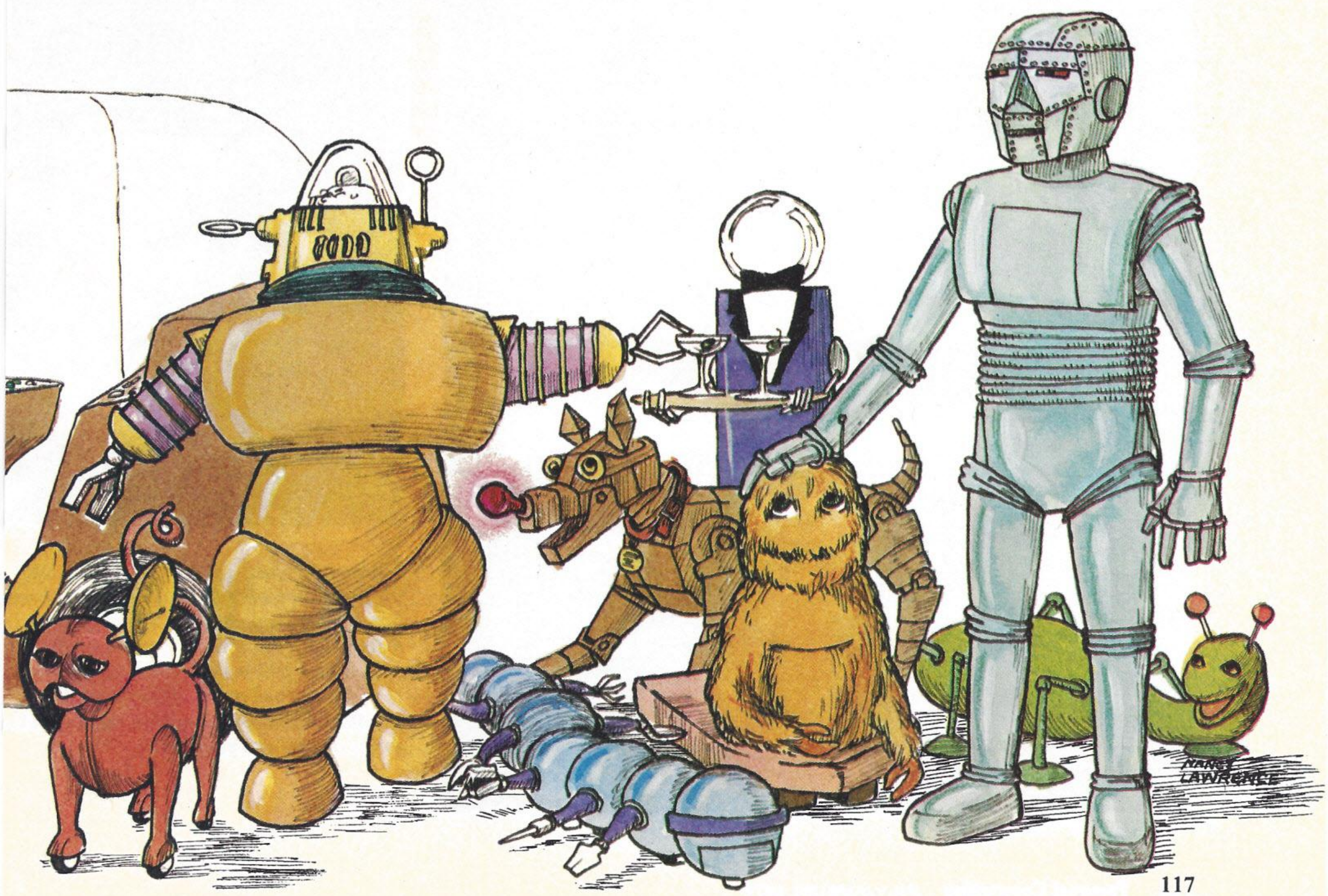
Indeed, it seems that the robots – bona-fide, true-to-life robots – are coming. We are on the brink of a technological revolution that represents the next quantum jump in the evolution of machines. From all available evidence it appears that amateur experimenters, rather than industrial or university researchers, are going to take the initiative.

The Delicate Formative Period

The formative period of any new technology is sensitive. Mistakes or misconceptions in the early going can confuse and misdirect the efforts of well-meaning experimenters for years to come. Unless we are careful about laying the basic philosophical foundations of robotics now, we run the risk of wasting time, effort and money developing machines and concepts that lead nowhere.

This is the time to get the basics of robotics straight; and the logical starting point seems to be working up a good definition of *robot.* Now that's really getting down to basics; but there is a need for carrying an initial analysis to that extreme of simplicity. The term *robot* is a coined expression that doesn't define itself as technical terms often do.

What many people think a robot should be really isn't a robot at all, and it is difficult for such people to understand the legitimate definition as they



try force-fitting it to their existing misconceptions. For that reason, it is perhaps a good idea to spell out first what a robot is *not* – to tear down some old structures and make way for a more useful and exciting one.

What a Robot Isn't

There are two major classes of electromechanical contrivances making something of a stir in the popular media these days. Some of them are quite complex and very interesting machines, but they are not real robots. They are merely imitations – *parabots,* if you will.

One class of parabot calls for having a human operator manipulate the machine by remote control. Would-be roboticists must be misled into believing any sort of remote-controlled machine that is manipulated by a human operator is any more vital to the evolution of machine technology than remote-controlled airplanes. Forget about any machine that relies on the on-line intervention of a human being.

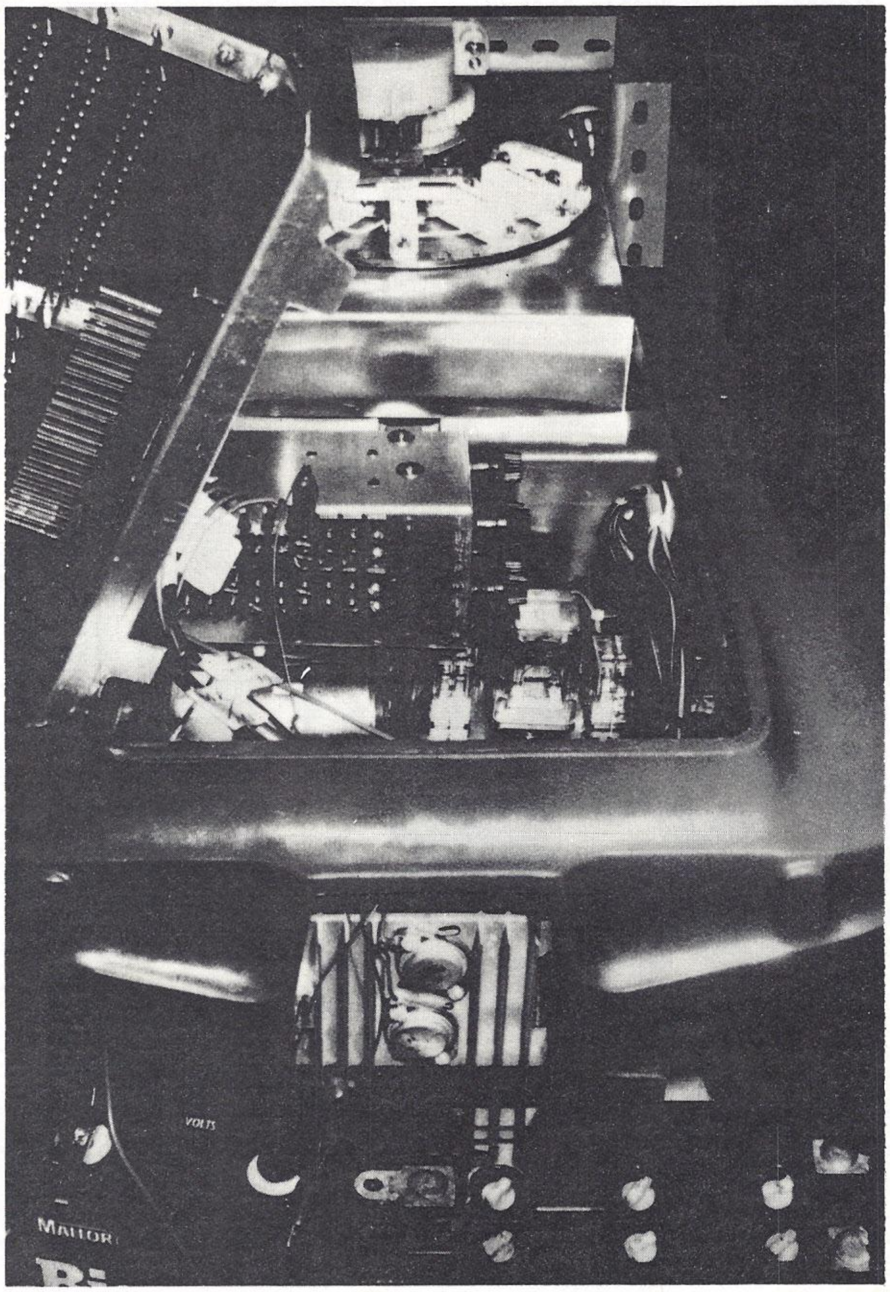
The second major class of parabots simply replaces the remote human operator with a small computer system. It is certainly possible to play an endless variety of sterile computer programs through a cleverly interfaced set of mechanical gadgets and end up with some fascinating effects. All this can be done, however, without really jumping the technological gap into the era of robotics.

ROBOT – A Matter of Semantics

When thinking about real robots, consider two alternate names, *cyborg* and *automaton.* These are the key expressions. The word *cyborg* comes from the same root as *cybernetics* – the science of closed-loop feedback or servo systems pioneered by Norbert Wiener in the 1940's. A robot, then, must have cybernetic features, but that only expresses one aspect of how the job is done. It doesn't really say what a robot is – what separates it from any other class of machine already in existence.

Now consider the term *automation.* This work comes from the same root as *automatic;* but what is even more meaningful is the fact it shares a common heritage with the word *autonomous* – and that word is the key to defining a robot.

A robot must be an autonomous machine; a machine capable of carrying out functions on its own. A typical computer system is not an autonomous machine. As sophisticated as some computers might be, they must interact with a human operator to do anything useful at all. A robot is not a slave, but a "free" machine. It is a free-will machine that can, indeed, obey



Photographs by Ron Dozer

Sherman Kennedy (above), lead technician for the author's Mark-II Buster robot development program, displays the latest results. Robot (left), Mark-II Buster, has a more streamlined power pack and logic system than the original model. When completed the Mod-4 version will exhibit more obvious forms of artificial intelligence.

the commands of a human operator, but only as long as those commands do not violate any higher-priority needs.

Given a command or goal by a human operator, a true robot must be free to execute that command and achieve the goal, freely deciding exactly how to go about it. And whenever the robot is not actively pursuing a goal set by its human operator, it must be free to determine and work toward goals of its own. This is not a flight of fantasy, but a prime example of what a robot – an autonomous cyborg – can and must do. Any machine incapable of exhibiting autonomous behavior is not a robot at all.

Integrative Behavior is the Key

The philosophy behind the construction of a truly autonomous cyborg, as incredible as the concept might seem at first, is not really difficult to implement these days. Buster III, described in *Build Your Own Working Robot,* is an example of a lower-order robot. Buster III can operate without the need for human intervention at all. He can seek out his own battery charger and feed himself when the need arises. He can work his way around most kinds of physical barriers and generally interact with his environment in a fashion that would clearly indicate some underlying intelligence.

Buster's brain is not a conglomeration of discrete, task-performing programs. The system is far more dynamic than is possible with the sort of thinking that goes into building parabots. The brain of a true robot is an integrated network of simple and basic functions that are orchestrated according to on-line environmental conditions and the task set before the machine.

A buster IV system, presently under construction, moves one step higher on the scale of robot technology. This new machine not only reacts in a quasi-rational manner to its environment, but has the capacity for learning how to deal with environment and even altering it if necessary and possible – *as judged by the machine!*

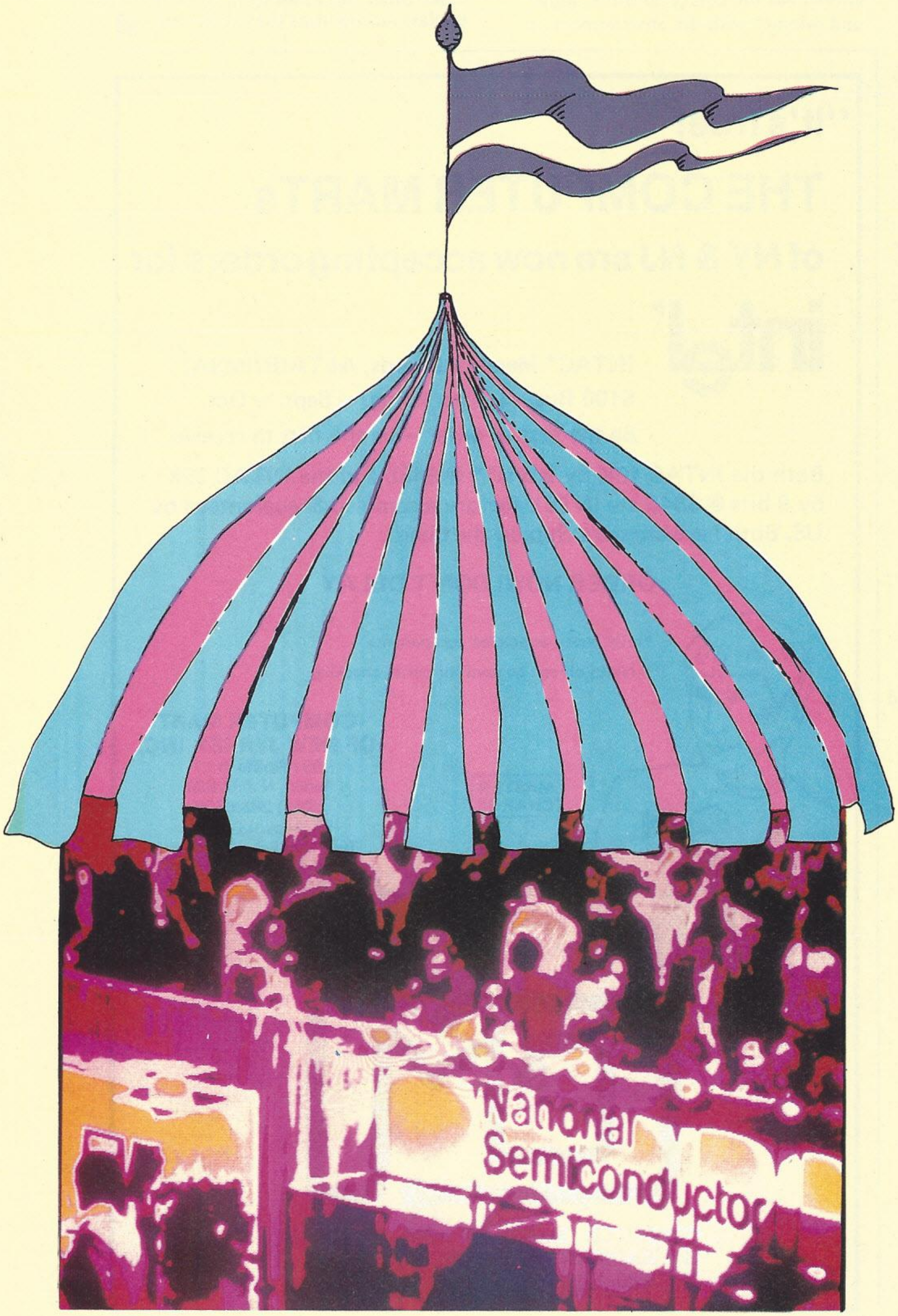
HOW More Important Than WHAT

Putting together the basic working definition of a robot and the integrative technique for implementing that definition, one central theme emerges: It is far more important at this point to think in terms of *how* a robot carries out its tasks than it is to become carried away with *what* it can or cannot do as a result.

What really distinguishes man from other animals? Of course we could point to an infinite variety of political, social, economic and technological achievements through the history of mankind, but all those things merely reflect something deeper in the human makeup. The real essence of man is bound up in how his mind works, rather than what he does as a result. Man is unique in his capacity for rational, imaginative and often highly abstract thought. No other animal has the ability to think, judge and interact with the environment on the same level man does, and so no other animal is capable of exhibiting such a high degree of achievement.

A true robot is to other machines as man is to other animals. If roboticists can shed their current misconceptions, and begin thinking in terms of an autonomous machine equipped with integrated reflex, decision-making and goal-setting mechanisms, we can expect to see a new order of machine exhibiting a rich variety of behavioral modes that make other machines seem to be the lower-class mechanisms they really are. ■

A FINE FAIRE



An ancient political stratagem is to hold all rallies in halls that are too small for the crowd. That way, the place is jammed; enthusiasts must be turned away at the door, and reports all indicate that the candidate was so beloved of his fellow citizens that he was almost crushed by admirers.

San Francisco's Civic Auditorium was jam-packed with mobs of personal computing enthusiasts for two days in mid-April when the well-publicized and well-produced Computer Faire opened. By the end of the second day, when impresario Jim Warren announced attendance at nearly thirteen thousand people, the event was being called "The *First Annual* Computer Faire." The event was a success and PERSONAL COMPUTING was glad to have been part of it.

Almost everybody who is anybody in the personal computing field was there, even some major companies like 3M, who have noticed the potential market for products they already make. Commodore, the calculator company that has been whispering cheerful predictions that they'll soon be producing a real computer for the mass market, continued to whisper. The company didn't have a booth of its own, but quietly displayed its prototype system in the booth of a calculator dealer, drawing quite a crowd.

If Commodore's PET computer is for real, at the price they project, the age of the consumer computer is almost here. They are hoping to sell a system with a calculator style alphanumeric keyboard (you'll have to be careful with your clumsy fingers), a video display, a computer complete with minimum 4K Ram and 12K ROM containing BASIC, and a cassette tape deck, all built – into a single cabinet – for a trifle under $500. There's some uncertainty about the price, of course, and talk about bigger and smaller models, but the proposition that a real company is really tackling the project is news.

Scoffers abound. The prototype was clearly a one-only developmental model, and insiders around gossipy Silicon Valley were busy complaining that Commodore couldn't pull it off before the end of the year. We shall see. The machine was percolating away very prettily. Other systems in this new class of consumer computers are ru-

mored, though none were shown.

Apple Computer Company brought a surprise to the show, the Apple II. The system is novel both in performance (BASIC in ROM, color video control built into the main board, etc. . .) and construction. The Apple II case is molded plastic of material and form suited for genuine mass production. The electronics in the system are remarkably simple, adaptable largely to automatic production techniques – and the system comes with two control units that allow players to use an array of game programs on any television set driven by the computer. The games and computers are in fact beginning to merge as consumer products.

It's impossible to list and discuss all of the important exhibitors who were there, best foot forward, showing their wares. To get the flavor and value of such a show, you must attend. Lots of good shows coming up. If you haven't caught one yet, don't despair.

Some sidelights – Sheldon Howard, of Micro Computer Devices in Montebello, California, had worked out a good promotional scheme in which he gave numbered, printed cards to everybody who came by. In fact, each number appeared on two cards and if one cardholder could find another with the same number, each got a small prize. As Howard hoped, people stuck these cards in their breast pockets, and displayed both the number and the company prominently.

Howard fell into a couple of traps (though without regret). Some of the cards issued were blank, without numbers. Various thoughtful Faire attendees went to him with matching blank cards, demanding and receiving their prizes. More to the point, Dan Meyer's bright kids put the computers to work in their dad's Southwest Tech booth. They scouted up all the cards they could find lying around the floor or in wastebaskets, and entered the numbers in the system as they came. The computer sorted the numbers, looking for matches. It found some, too, and the bemused Howard paid off.

PERSONAL COMPUTING's staff, exhausted from passing out several thousand sample copies of the magazine in two days, discovered that a hundred extra cartons of magazines remained to be shipped out at the end of the show. Careful planning had provided only half a dozen address labels to slap on the cartons and volunteers for writing out ninety-four more labels by hand did not step forward.

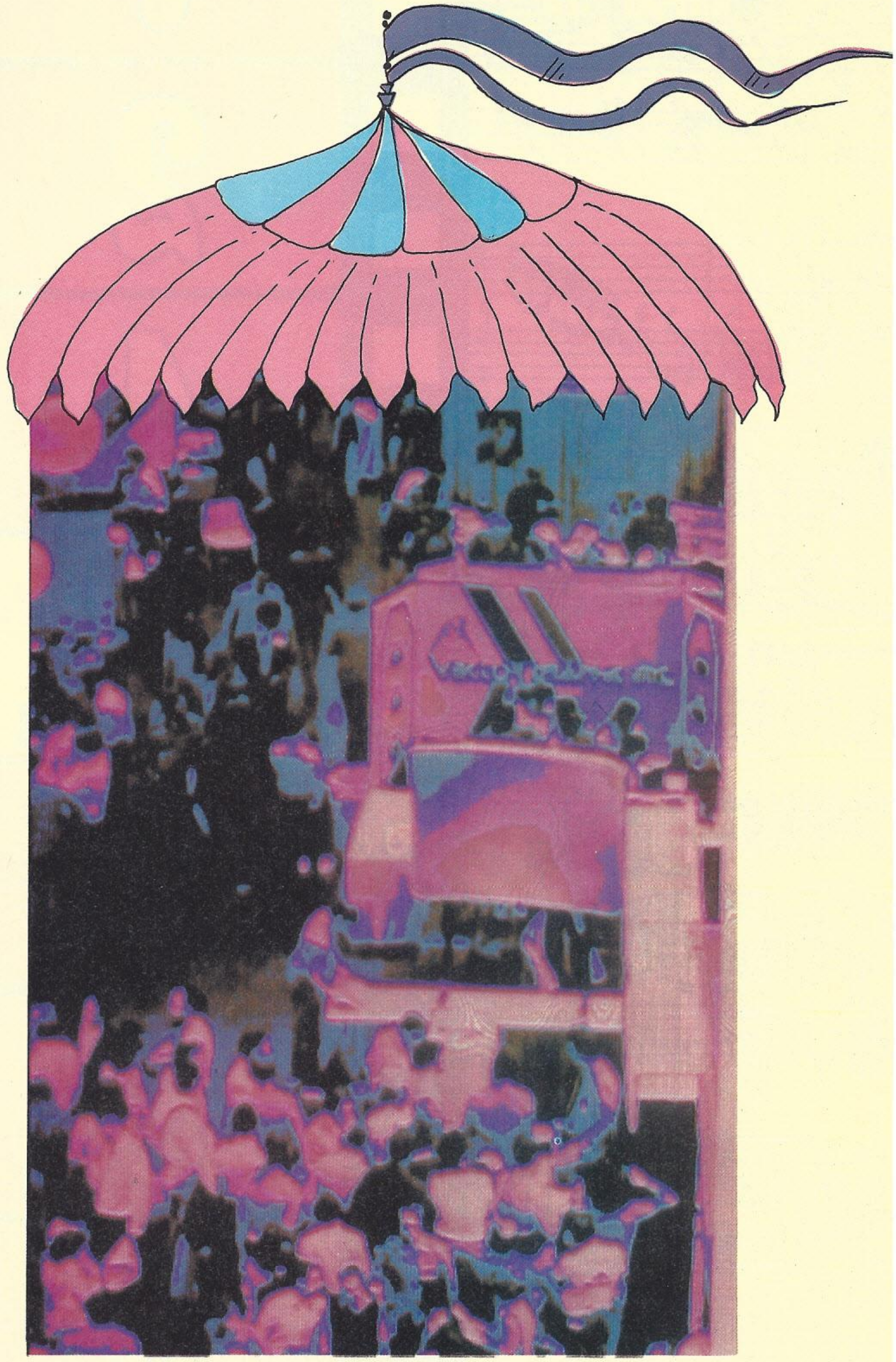
Luckily, the Digital Group booth was still operating and their nifty new matrix printer was hooked up, set to run. They typed in the address and told the computer to make a hundred copies. Amazing! We always suspected computers were good for some thing.

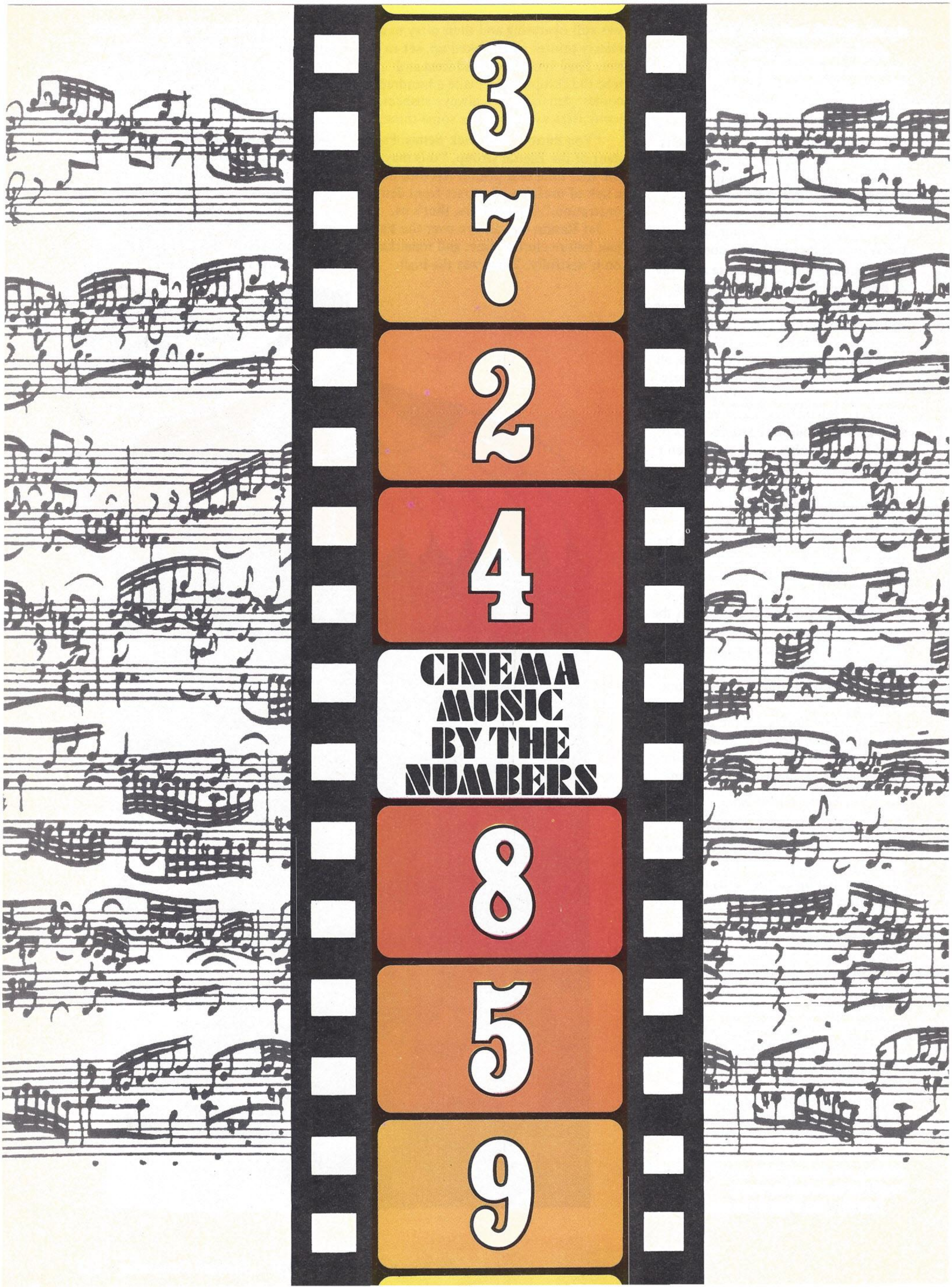
"You know," said Dick Bemis, President of the Digital Group, "this may be the first time that printer has ever done a lick of useful work, apart from demonstration." Trail-blazers, that's us.

Jef Raskin cast an eye over the Faire just before closing time, and remarked on it wistfully. "This was the high point of something," he said. "I don't know how the excitement of pioneering can be preserved. After this, whatever happens will be very different."

He's right. The fun will continue, and excited newcomers will do their own pioneering, never looking back to the scuffling and fellowship of the first couple of years of personal/hobby computing.

Big things are coming. This Computer Faire looked both forward and back with good spirit. It was well-planned; the hall was a bit too small. ■




3
7
2
4
CINEMA
MUSIC
BY THE
NUMBERS
8
5
9

FUTURE COMPUTING

by Jean Renard

Motion picture production is a craft that rarely rises to "art." However, the better the craftsman, the more likely he is to become an artist. In the next few years, personal computers will develop into craftsman's tools that greatly improve the possibility of producing art, even in the workshops where we labor daily to produce as much as we can with as small a budget as possible.

In one area, the application of computers seems obvious for the filmmaker, and great strides are already being made. If we look intently into the future, we can see the general outlines of big things. The subject of interest is music for the cinema.

Music Serves Many Functions

Almost every film produced – for training, sales, public relations, propaganda, entertainment – uses music. Music does not merely establish and reinforce mood by sounding romantic during a love scene or threatening just before some violent occurrence; music also ties sequences of action together, smoothing what might otherwise seem like a choppy, uncertainly-related series of images. Even more prosaically, the music that comes up behind the opening titles of a film helps the projector operator to adjust the level of sound for the room. It prepares the audience for the quality and volume of the sound track. Notice that the few motion pictures without opening music are often confusing at first. When someone speaks in the film, either a character performing or the voice-over narrator, the first sentence is unintelligible, and its meaning is lost. You don't know yet what you are listening for and the opening phrases are lost while you adjust to the situation.

Even the duration of the opening music is often determined by a rule-of-thumb. For example, an audience can read a simple title on the screen in about six seconds. Suppose that this article were the script of a commercial film; it might begin as shown in the chart below.

PICTURE	AUDIO
Title: Cinema Music by the numbers	1 **Music**
Title: by Jean Renard	2 **Music**
Dolly shot as camera moves through working production studio with lights, set, etc. . . while crew is intent on its business'	3 (Music under as voice-over comes in) **Narrator:** Motion picture production is a craft that rarely rises to "art." However, the better the craftsman, the more likely he is to become an artist.
Camera holds on two-shot of director and camera operator. Operator is leaning aside as director looks through a camera viewfinder. Director straightens up and gestures toward the set, striking a thoughtful attitude.	4 (Music fades down and under as "live" studio sound comes up behind narration.) **Narrator:** In the next few years, personal computers will develop into craftsmen's tools that improve the possibility of producing art. . . . .

Putting It Together

Now, the creation of this scene is a simple matter of putting the elements of motion picture production down on paper. It's done by the numbers. Even the scenes are numbered, shot by shot, so that the director, editor and post-production technicians can refer easily to the materials with which they're working, picking their way repeatedly through the script for different purposes. (Scripts are prepared in various forms, but this one is typical.) The innovative user of personal computers can probably already see half a dozen applications of his computer to the mechanical aspect of film production, but let us consider only the music in this essay.

The editor of the film, cutting sound track and picture together, will look at the scene marked "1" and realize that the audience is being presented with a short title. He will physically cut a piece of film that is coated with magnetic oxide, just like that on quarter-inch recording tape, and splice it into a reel of sound materials that he is creating right beside the reel of pictures. The piece of mag film he selects has been recorded with a piece of suitable opening music.

The editor lets that run until about three-and-a-half-feet of the sound film has unreeled from the beginning. If he is working in 16mm film, this will give him almost six seconds of music. He can hear the music, because the mag film is being pulled over an ordinary tape recorder pickup head connected to an amplifier and a tiny speaker.

When he gets almost six seconds of music reeled out, he begins to listen for a good break point in the music, the end of a bar, a change in tempo, some audio marker that will nicely match a cut in the picture from the first title to the second. He finds such a marker in the music track and then cuts his picture to match. *(continued next page)*

That's the way it goes, through the whole film, if this is an ordinary little commercial/industrial production. Strictly by the numbers, with a bit of judgement applied according to the craft, or art, of the editor. Since the mag film is provided with sprocket holes, just like those in the picture film, the sound and picture reels can be synchronized through all subsequent production steps merely by locking both pieces of film mechanically into matching sprockets on a single shaft. Again, by the numbers. This is how speech is synchronized with mouth movements in the pictures, while complete freedom is maintained by the editor in cutting sound and picture to suit himself.

Music From Where?

First, where does the music come from? Most of it is available in the form of record collections, music specially recorded for motion picture production use. The music in these libraries has been written for film, with good, clean beats, clear changes in tempo and key, so that the editor *can* find distinct audio markers. Often, the music on a disc, many separate cuts of the same theme, for example, performed in different moods and styles, is recorded in the same key. This allows any cut to be matched with any other, keeping jarring changes to a minimum.

A "music editor" may look over the script of a film, talk to the director, and look at some of the footage, then pick selections from the record library. He has the selections transferred to mag film, then sits down to score the picture, if music is to be used throughout, and if the budget permits. (More often, the producer adds head-and-tail music to a track without any formalities after the narration and pictures have been edited together.)

The music editor may cut fifty separate passages of music for a score, working back and forth, trying this, trying that, judging the whole thing from that tinny speaker. The music on his mag film is all recorded at the same level, without fades and dissolves, all fairly loud. The adjustments in volume aren't made until the music track is mixed with the voice track and the effects track, if any, late in the process. Cut and paste, by the numbers.

When Control Is Complete

In the case of animation, where the pictures are drawn frame-by-frame, and the director has absolute control of the timing, the sound track is usually recorded and roughly edited *first.* In this case, the action can be made to fit the music exactly.

The selected music is transferred first to mag film (usually 35mm for animation – why not go first class when everything else costs so much in animation?), and the director does frame-by-frame "measurements" of the music. Often a director may be seen standing over a motor-driven machine that pulls the mag film under the playback head at standard projection speed. The director listens to the music carefully, tapping out the beat with a grease pencil. The pencil makes marks right on the mag film, next to the pickup head.

For example, he may tap with each downbeat of the music, using an orange grease pencil. After the film has run all the way through, each downbeat is marked. He may then go back and mark each upbeat with a white grease pencil.

After this marking, he can sit down with the film and determine exactly what's happening musically at each foot and each frame of the production. From this data, he prepares an "exposure sheet" (Fig 1). That is, he marks a piece of paper that is printed with lines and numbers indicating frames and feet, to record the data so an artist can use it to determine what pictures must be drawn where. (The exposure sheet also shows which speech sounds occur at which frames and where the sound effects are to be placed.) By the numbers.

Without Control

Imagine that the producer has photographed and edited a documentary film on some primitive tribe, the timing of whose actions he could not control. Now he wants to have a composer write original music for the score. He may give the composer an exposure sheet derived from the pictures so that the music can be created to fit. When the orchestra records this music, the session is dominated as much by technical concerns as by artistic concerns. The conductor must know how fast to play each passage and may actually work with the projected motion pictures . . . at great expense. More often, the conductor works by the numbers. The composer has used the exposure sheet to determine the rate at which each passage must be played, and he specifies for each passage a number that means something very specific. That number determines the speed of a metronome to which the conductor pays strict attention. Again, the score is created piecemeal, assembled with the other elements of the complete motion picture later. (True, the motion picture making process is a bit more specialized than indicated here, with "click tracks" instead of conventional metronomes to make the music fit the mechanics of film production easily, but the explanation of the mechanics is tedious.)

The producer pays for all of this in time and money, whether he is merely

Frame Foot	Voice	Music	Effects	Level One	Level Two	Lev
96	DOG H	X		HEAD BACK	BODY AND	
97	e			HOWLING	ARMS RELAXED	
98	e			WITH QUAVER		
99	e			TO MATCH		
100	e			MUSIC		
101	L					
102	L					
103	L					
104	L					
105	P					
106						
107	H	X				
108	e					
109	e					
110	e					
111	L					
112	L					
	L					
	P					

Figure 1

renting the use of the music from the owners of the rights to the recorded library or is buying all rights to an original score under the supervision of the musician's union and ASCAP, the organization that keeps track of royalty payments. Cost is high.

Computers Are Here

Now, for the future application of computers.

Considering the frantically rapid development now of digital music synthesizers with all of the basic attributes of computers – and some of the attributes of real musical instruments – we can see a time when the exposure sheet is created by a computer system which then goes on to create the completed score, actually performing and recording the music.

The fascinating aspect of this is not in the obvious ability of a skilled operator with a computer to work by the numbers in fitting music library selections to a film while handling all adjustments in tempo, synchronism, duration and sound level. Of course a computer system is perfectly adapted to such chores, and systems are already being modified and programmed to do the work. (No commercial service offered by a Lemonade entrepreneur has yet come to our attention, but it's only a matter of time.)

No, the excitement is found in the realization that the computers will be able to *compose* music to specification. Do not stop your imagination at the point of setting the computer loose with an exposure sheet to create "musical sounds" with the right timings to match the action, and in the right key. That's useful, but trivial.

Music Appreciation

Imagine instead that some big computer system has been set to "listening to" a large library of conventional music. Let us say that the works of Scarlatti, Bach, Thelonius Monk, Dave Brubeck, MJQ, Ferde Grofe, and the Beatles among others have been played repeatedly to the big system. Each selection of music is accompanied by comments – the name of the composer, the name of the selection, the date of composition, the personnel of the recording group, the instruments employed, the nature of the hall in which the recording was made, perhaps the nature of the recording and playback systems. The comments include the reports of the critics on the works. Some music editor supplied comments that describe the works in terms useful to motion picture producers . . . "silly. . . suspense-full . . . dramatic . . . psycho . . . soaring.

The big system listens to the music and the comments, and processes the information through a pattern recognition routine. Mind you, the nature of this pattern recognition routine is not yet clear to anyone. It is clear only that the computer must be able to associate terms like "suspenseful" not only with particular pieces of music, but with particular *classes* of music. The programmers of this system may never actually know what cues the computer uses to make its associations.

Making New Music

After this learning mode, when all of the computer's impressions of music have been stored in digital form, the music editor should be able to give the computer an exposure sheet specifying, by the numbers, what sort of music is needed at each point in the film, at what volume.

For example, "from frame 14,400 (exactly ten minutes into the film) to frame 15,840 (exactly one minute later), I want a bouncy waltz that sounds like a Beatles-style satirical takeoff of a Scarlatti harpsichord sonata that blends midway through the passage into a performance of the dixieland Firehouse Five plus Two. Hit tempo and key changes within a frame of frames 14,982 and 15,320."

The computer has a virtue that live composers do not; the machine will not be offended and disgusted by what you ask. It will reach into its impressions of music and comments to find elements that it can combine according to legitimate, rigorous musical disciplines to synthesize what you ask for.

The computer can jostle and shift the elements of the music to fit your specified cues. If your specifications are unrealistic, the computer will do its best, and provide you a hodgepodge of sound that meets the letter of your request, if not its intent. If you are a shrewd judge of music and have learned something of the computer's tastes and impressions, you can work gracefully with the system to produce the sound that you want.

If the output is an analog recording on mag film, you can synchronize your score directly with the pictures to enjoy the results.

The producer can modify the finished work to his heart's content, imposing his own judgements on the score, so that it is molded gradually into something that does the intended job. The computer becomes a craftsman's tool.

Big And Small Systems

Since time and core size are both major considerations in all contemplated or currently experimental pattern recognition schemes, the "learning" computer must surely be a large one, beyond the scale of any envisioned personal system. However, the recorded "impressions" need not exceed the capacity of present digital disc systems. It seems entirely plausible that personal computers can, within a few years, employ the recorded data for references that will fill the request of the user for music with certain characteristics. The small computer can then instruct a good synthesizer on what to synthesize. Notice that the system is not merely a plagiarist, swiping a few bars here and a few there for a combination in a monument to theft. Consider that anyone who takes the time can learn to recognize the work of a particular composer, a particular performer, even a particular conductor, merely by catching a few bars or even a few notes of a piece being played. Often, the recognition is accurate even when the piece of music is new to the listener. The listener "just knows" whose work he is hearing.

Assuming that this knowledge is not derived mystically, but is part and parcel of the same physical arrangement of phenomena that the computer will be exposed to, we can be wholly confident that the computer will "just know" what to synthesize. You want a saxophone piece that sounds like something Brubeck wrote to emulate a Milhaud pastorale being played by Paul Desmond? Why not?

Your local Lemonade Computer Service Company, drawing on a purchased set of impressions, should be able to score a film with wholly original, non-electronic-sounding music (and effects? and voices?) rapidly, at modest cost. By the numbers – modified by the good, craftsmanlike, artistic judgement of the human being who is spending time, money and talent on the production.

This reality is growing ever clearer through the mist that screens us, mercifully, from the future. Skilled, dedicated amateurs, who can spend their time in experiments, are creating this reality. Some of us hope for very rapid progress.

On the other hand, as one Hollywood music editor commented, "Good Lord! The musician's union will absolutely come unglued!" Yes, so will ASCAP . . . and the rest of us who struggle to adapt in a changing, but interesting world. Surely, this is part of the future of computing. ■

Classifieds

Personal Computing Classified Rates

Individual: $1 per word; address set free; first two words set in all capital letters for free. If you want **bold** type, add $1 for each word set bold. Add $5 if you want replies sent to us and forwarded to you.

Company: 1X: $60/inch; 3X: $55/inch; 6X: $50/inch. Display ads accepted only if camera-ready materials submitted. Agency commission paid on camera-ready ads and after the first insertion of a publisher-set ad if subsequent insertions run without changes. Publisher sets ads in format and attempts to fit copy into space requested.

Advertisement accepted, however, with understanding that space may be larger than requested. Minimum billing size one inch; ads billed in half-inch increments thereafter. Frequency rate applies to publisher-set ads only if they run without changes in subsequent issues.

To place an ad, send your copy to Classified Ad Dept., **Personal Computing,** *167 Corey Rd., Brookline, MA 02146.*

Personal Computing

ADVERTISERS' INDEX

ADVERTISING SALES OFFICES

Northeast: George Palken, 167 Corey Rd., Brookline, MA 02146; (617) 232-5470; Arthur Daks, Benwill Publishing Corp., 299 Madison Ave., N.Y., N.Y. 10017; (201) 661-0360 • **Mid-Atlantic:** Robert Poggi, 69 So. Central Ave., Valley Stream, NY 11580; (516) 872-9550 • **Mid-West;** Ralph Petersen, 1 Wheaton Ctr. No. 1706, Wheaton, IL 60187; (312) 653-2480 • **Southwest:** David Bunnell, 401 Louisiana, S.E. No. G, Albuquerque, NM 87108; (505) 266-1173 • **West:** Yuri Spiro & Carol Stagg, 823 Enchanted Way, Pacific Palisades, CA 90272; (213) 454-0624

If you're moving...

Please allow at least six weeks for us to put your new address into our subscription computer. Send your new address and the mailing label with your old address to: Circulation Dept., Personal Computing, 167 Corey Rd., Brookline, MA 02146.

Or use this convenient form.

Please attach here the mailing label with your old address

Write in here your new address.

NAME ______________________________

ADDRESS ______________________________

CITY ____________ STATE ____________ ZIP ______

Artist and Computer

A Review by Jef Raskin

There have been a number of toys on the market that produce rather attractive graphics. One works by a number of interlocking planetary gears, another by whirling a platter on which viscous paints have been placed, a third is built of a pair of crossed pendulums and produces damped Lissajous figures. If done on good paper and properly framed no doubt these works could pass as computer art. Showing them as art would be considered "camp" and stand as a fine example of *chutzpah,* but then art is mostly an exercise in public relations for a reasonably large segment of the artists in our culture. The idea of computers in art is *per se* captivating, never mind the quality of the work. The media cannot resist so popular a topic. This book has as much critical stance as a daily tabloid. A few good works do inevitably creep in, but this is due to the law of averages, not the perspicacity of the author. In general, the works have the quality of being produced by artistic toys.

The book itself consists of a sequence of essays by the artists with excellent reproductions of their work. It is well laid out, looking very much like an art book, with sans serif typography and lots of space. The color plates are good. The essays remind us that being an artist does not necessarily make one articulate about one's work or vice versa.

Many of the artists represented in the book frankly admit that they do not know how to program. The significance in this is that there is an unseen (and often unsung) person (or persons) who may be guiding the work as much as is the official "artist." I have no quarrel with this procedure except to note that in this case the artist would be more honest to call the work a "found object." Attribution of a work, since Duchamp, is to the person who declares a thing to be art and not to the person who created it for some other purpose.

But if the programmer knew that the output was intended to be art we have a new situation. The "artist" is really a *patron.* A patron is one who desires that a work be created, gives a loose description of the work, and the artist produces it. The patron may reject a work, and ask for changes but this still leaves the impression that the wielder of the artistic materials (here called a *programmer*) may well have been the actual creator of the work and should be given due credit.

Technologists who have experimented with electronics and computers extensively will find many of the works rather familiar. To someone coming from a strict art background they might well appear attractive. But, as with the graphics produced by the toys, the interest quickly palls. Given enough time and exposure, the readily made and, for the most part simple minded, graphics will disappear from the galleries and join paintings of wide-eyed children in the interior decorating sections of our large department stores.

With few exceptions the reader is informed that the works were produced, for example, on a "Superstar Seven Thousand computer with the Spangled operating system, driving a Blotto Plotter over a phone line at 2400 baud." But almost all the articles ignore the *algorithms* that really produced the works. The frequent citation of computer and language employed shows that few of the people involved realized that they are generally *irrelevant*. The same algorithm written in some other language on some other computer would produce an *identical* work. If any details are to be published, a flow chart of the algorithm used would be the obvious first choice. Giving the computer's make and model is somewhat akin to specifying the wood of which Rembrandt's easel was made. Useful trivia, no doubt, for crossword puzzles.

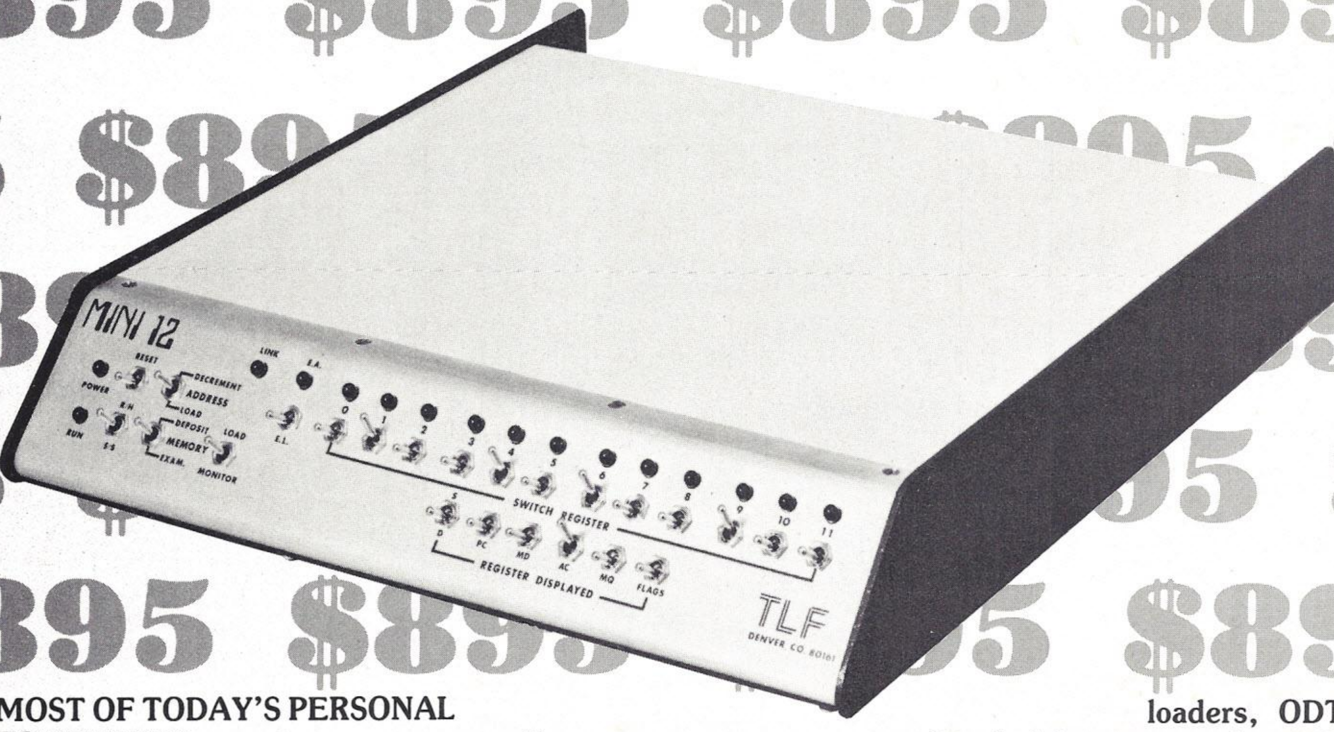
Artist and Computer substantiates the hypothesis that computer art is weak art. Its strength rests on the fascination that the public has with its misimpressions about computers. As that myth fades, so will books like this.

This simple algorithm, here encoded in BASIC, produces 100 works of art. The instruction DARKEN X, Y means merely to color in a square at those coordinates. Readers with plotters will be able to write a subroutine at that point that will do it automatically. The reviewer guarantees that these works will be acceptable as computer art anywhere in the world. So long as the viewer hasn't read this review.

```
FOR  N  =  40  TO  140
FOR  X  =  0  TO  127
FOR  Y  =  0  TO  47  STEP  INT  (X/N+1)
DARKEN  X,Y
NEXT  Y
NEXT  X
PRINT  "GO  ON  TO  NEXT  WORK  OF  ART"
NEXT  N
STOP
```

The INT means the integer part of the following expression. For proper effect it should be done on fine paper with india ink, or very large with acrylics in random colors on canvas. Frame it nicely. You are now a computer artist. The adventuresome type will try varying some of the constants in the program. Then you will be creating really *original* works of art. Many of the artists in the book have done no more than that. ■

Why buy a micro, when you can buy a mini for less!



MOST OF TODAY'S PERSONAL COMPUTERS are microcomputers. Compared to our minicomputer they are slow and expensive.

Our mini "the MINI 12™" has been around for three years. We've been steadily selling it to industrial customers who have proven its reliability. Only recently have we paid much attention to the personal computing market. We've discovered that our computer is simply better, and most interestingly, LESS EXPENSIVE.

NONE OF THE MICROCOMPUTERS can offer you the range of software available to the MINI 12. Compatibility with the most popular minicomputer, DEC's PDP-8E, gives you a vast library of programs. You will also be pleasantly surprised when you experience the speed and sophistication of our EXTENDED BASIC compiler, or FORTRAN, ALGOL and FOCAL.

Unlike today's microcomputers, our Mini 12 is fully software supported by an advanced operating system, including...file handling text editor, symbolic assembler, loaders, ODT (octal debugging technique), batch processor, and system utilities.

The MINI 12 comes with 8K of memory and is easily expandable to 32K. Because the MINI 12 processes 12 bit words (instead of 8 bit bytes) you need far less memory. For most applications... using our optional MINI STORE digital cassette system with the BASIC compiler, you will never need add another single byte!

Because we want you to have the pleasure of using your minicomputer right away, our MINI 12 comes fully assembled, debugged and ready to plug in. It includes a full front panel, controller for eight digital tape drives, and built-in serial interface plus other technical niceties like a binary loader and monitor bootstrap in ROM memory.

Frankly, when compared with microcomputers, we're surprised at how good a deal we are. Write... or better yet, give us a call @303-794-1634 and let us help you get it all together!

TLF Post Office Box 2298, Littleton, Colorado 80161, Telephone 303-794-1634